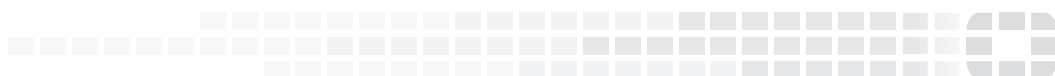




**FORTINET**  
*High Performance Network Security*



# FortiOS™ Handbook - VoIP Solutions: SIP

**VERSION 5.6.0**



## **FORTINET DOCUMENT LIBRARY**

<http://docs.fortinet.com>

## **FORTINET VIDEO GUIDE**

<http://video.fortinet.com>

## **FORTINET BLOG**

<https://blog.fortinet.com>

## **CUSTOMER SERVICE & SUPPORT**

<https://support.fortinet.com>

<http://cookbook.fortinet.com/how-to-work-with-fortinet-support/>

## **FORTIGATE COOKBOOK**

<http://cookbook.fortinet.com>

## **FORTINET TRAINING SERVICES**

<http://www.fortinet.com/training>

## **FORTIGUARD CENTER**

<http://www.fortiguard.com>

## **FORTICAST**

<http://forticast.fortinet.com>

## **CLI REFERENCE**

<http://cli.fortinet.com>

## **END USER LICENSE AGREEMENT**

<http://www.fortinet.com/doc/legal/EULA.pdf>

## **FEEDBACK**

Email: [techdocs@fortinet.com](mailto:techdocs@fortinet.com)



Thursday, March 30, 2017

FortiOS™ Handbook - VoIP Solutions: SIP

01-560-99686-20170330

# TABLE OF CONTENTS

|  |           |
|--|-----------|
| <b>Change Log</b>  | <b>7</b>  |
| <b>Introduction</b>  | <b>8</b>  |
| Before you begin   | 8         |
| How this guide is organized  | 8         |
| FortiOS 5.6 VoIP new features  | 9         |
| SIP strict-register enabled by default in VoIP Profiles (380830)     | 9         |
| SIP diagnose command improvements (376853)                           | 10        |
| <b>Inside FortiOS: Voice over IP (VoIP) Protection</b>               | <b>11</b> |
| Advanced Voice over IP Protection                                    | 11        |
| Deep SIP message inspection (also called deep SIP header inspection) | 11        |
| SIP message rate limiting  | 11        |
| RTP and RTCP pinholing   | 12        |
| Stateful SIP dialog tracking   | 12        |
| Inspecting SIP over SSL/TLS (secure SIP)                             | 12        |
| Inspecting SIP on multiple ports                                     | 13        |
| Carrier Grade protection   | 13        |
| High Availability  | 13        |
| Geographical Redundancy of SIP servers                               | 13        |
| Logging and Reporting  | 13        |
| NAT/NAPT   | 13        |
| Header Manipulation  | 13        |
| Hosted NAT traversal (HNT)   | 14        |
| IP Address conservation for NAT                                      | 15        |
| SIP ALG activation   | 15        |
| SIP over IPv6  | 16        |
| Platform support and hardware acceleration                           | 16        |
| <b>Common SIP VoIP configurations</b>                                | <b>18</b> |
| Peer to peer configuration   | 18        |
| SIP proxy server configuration                                       | 18        |
| SIP redirect server configuration                                    | 19        |
| SIP registrar configuration  | 20        |
| SIP with a FortiGate   | 21        |
| <b>SIP messages and media protocols</b>                              | <b>24</b> |
| Hardware accelerated RTP processing                                  | 26        |

|   |           |
|---|-----------|
| SIP request messages .....  | 26        |
| SIP response messages .....   | 27        |
| Informational (or provisional) .....  | 27        |
| Success .....   | 28        |
| Redirection .....   | 28        |
| Client error .....  | 28        |
| Server error .....  | 28        |
| Global failure .....  | 29        |
| SIP message start line .....  | 29        |
| SIP headers .....   | 29        |
| The SIP message body and SDP session profiles .....                                   | 32        |
| Example SIP messages .....  | 34        |
| <b>The SIP session helper .....</b>   | <b>36</b> |
| SIP session helper configuration overview .....                                       | 36        |
| Disabling and enabling the SIP session helper .....                                   | 37        |
| Changing the port numbers that the SIP session helper listens on .....                | 38        |
| Configuration example: SIP session helper in Transparent Mode .....                   | 38        |
| General configuration steps .....   | 39        |
| Configuration steps - GUI .....   | 39        |
| Configuration steps - CLI .....   | 40        |
| SIP session helper diagnose commands .....  | 41        |
| <b>The SIP ALG .....</b>  | <b>43</b> |
| Enabling VoIP support from the GUI .....  | 45        |
| SIP ALG configuration overview .....  | 45        |
| VoIP profiles .....   | 46        |
| Changing the port numbers that the SIP ALG listens on .....                           | 47        |
| Disabling the SIP ALG in a VoIP profile .....   | 47        |
| SIP ALG diagnose commands .....   | 47        |
| <b>Conflicts between the SIP ALG and the session helper .....</b>                     | <b>49</b> |
| <b>Stateful SIP tracking, call termination, and session inactivity timeout .....</b>  | <b>50</b> |
| Adding a media stream timeout for SIP calls .....                                     | 50        |
| Adding an idle dialog setting for SIP calls .....                                     | 51        |
| Changing how long to wait for call setup to complete .....                            | 51        |
| <b>SIP and RTP/RTCP .....</b>   | <b>53</b> |
| <b>How the SIP ALG creates RTP pinholes .....</b>                                     | <b>54</b> |
| <b>Configuration example: SIP in Transparent Mode .....</b>                           | <b>56</b> |
| General configuration steps .....   | 56        |
| Configuration steps - GUI .....   | 56        |
| Configuration steps - CLI .....   | 58        |
| <b>RTP enable/disable (RTP bypass) .....</b>  | <b>59</b> |
| <b>Opening and closing SIP register, contact, via and record-route pinholes .....</b> | <b>60</b> |

|   |           |
|---|-----------|
| <b>Accepting SIP register responses</b>   | <b>61</b> |
| <b>How the SIP ALG performs NAT</b>   | <b>62</b> |
| SIP ALG source address translation  | 62        |
| SIP ALG destination address translation   | 63        |
| SIP Call Re-invite messages   | 63        |
| How the SIP ALG translates IP addresses in SIP headers                                    | 63        |
| Source NAT translation of IP addresses in SIP messages                                    | 64        |
| Destination NAT translation of IP addresses in SIP messages                               | 65        |
| How the SIP ALG translates IP addresses in the SIP body                                   | 66        |
| SIP NAT scenario: source address translation (source NAT)                                 | 67        |
| SIP NAT scenario: destination address translation (destination NAT)                       | 70        |
| SIP NAT configuration example: source address translation (source NAT)                    | 73        |
| Configuration steps - GUI   | 73        |
| Configuration steps - CLI   | 75        |
| SIP NAT configuration example: destination address translation (destination NAT)          | 76        |
| General configuration steps   | 76        |
| Configuration steps - GUI   | 76        |
| Configuration steps - CLI   | 78        |
| SIP and RTP source NAT  | 79        |
| SIP and RTP destination NAT   | 80        |
| Source NAT with an IP pool  | 81        |
| Different source and destination NAT for SIP and RTP                                      | 81        |
| NAT with IP address conservation  | 82        |
| Configuring SIP IP address conservation for the SIP ALG                                   | 83        |
| Configuring SIP IP address conservation for the SIP session helper                        | 83        |
| Controlling how the SIP ALG NATs SIP contact header line addresses                        | 83        |
| Controlling NAT for addresses in SDP lines  | 84        |
| Translating SIP session destination ports   | 84        |
| Translating SIP sessions to a different destination port                                  | 84        |
| Translating SIP sessions to multiple destination ports                                    | 86        |
| Adding the original IP address and port to the SIP message header after NAT               | 87        |
| <b>Enhancing SIP pinhole security</b>   | <b>88</b> |
| <b>Hosted NAT traversal</b>   | <b>91</b> |
| Configuration example: Hosted NAT traversal for calls between SIP Phone A and SIP Phone B | 92        |
| General configuration steps   | 93        |
| Configuration steps - GUI   | 93        |
| Configuration steps - CLI   | 94        |
| Hosted NAT traversal for calls between SIP Phone A and SIP Phone C                        | 96        |
| Restricting the RTP source IP   | 96        |
| <b>SIP over IPv6</b>  | <b>97</b> |
| <b>Deep SIP message inspection</b>  | <b>98</b> |

|   |            |
|---|------------|
| Actions taken when a malformed message line is found .....              | 99         |
| Logging and statistics .....  | 99         |
| Deep SIP message inspection best practices .....                        | 100        |
| Configuring deep SIP message inspection .....                           | 100        |
| Discarding SIP messages with some malformed header and body lines ..... | 102        |
| Discarding SIP messages with an unknown SIP message type .....          | 102        |
| Discarding SIP messages that exceed a message size .....                | 102        |
| Discarding SIP messages with lines longer than 500 characters .....     | 103        |
| <b>Blocking SIP request messages .....</b>                              | <b>104</b> |
| <b>SIP rate limiting .....</b>  | <b>106</b> |
| Limiting the number of SIP dialogs accepted by a security policy .....  | 108        |
| <b>SIP logging .....</b>  | <b>109</b> |
| <b>Inspecting SIP over SSL/TLS (secure SIP) .....</b>                   | <b>110</b> |
| Adding the SIP server and client certificates .....                     | 112        |
| Adding SIP over SSL/TLS support to a VoIP profile .....                 | 112        |
| <b>SIP and HA-session failover and geographic redundancy .....</b>      | <b>114</b> |
| SIP geographic redundancy .....   | 114        |
| Supporting geographic redundancy when blocking OPTIONS messages .....   | 115        |
| Support for RFC 2543-compliant branch parameters .....                  | 116        |
| <b>SIP and IPS .....</b>  | <b>117</b> |
| <b>SIP debugging .....</b>  | <b>118</b> |
| SIP debug log format .....  | 118        |
| SIP-proxy filter per VDOM .....   | 119        |
| SIP-proxy filter command .....  | 119        |
| SIP debug setting .....   | 119        |
| Display SIP rate-limit data .....                                       | 120        |

## Change Log

| Date           | Change Description |
|----------------|--------------------|
| March 30, 2017 | Initial release.   |

# Introduction

This FortiOS Handbook chapter contains detailed information about how FortiGates processes SIP VoIP calls and how to configure the FortiGate to apply security features to SIP calls. This document describes all FortiGate SIP configuration options and contains detailed configuration examples.



This document uses numeric IP addresses for all SIP end points. SIP addresses can also use domain names instead of addresses. For the example, the following SIP addresses could refer to the same SIP end point:

```
inviter@10.31.101.20  
inviter@example.com
```

## Before you begin

Before you begin to configure VoIP security profiles, including SIP, from the GUI you should go to **System > Feature Select** and turn on **VoIP** (under **Additional Features**).

Also, VoIP settings are only available if the FortiGate or current VDOM **Inspection Mode** is set to **Proxy**. To view the inspection mode go to **System > Settings** to confirm that **Inspection Mode** is set to **Proxy**. You can also use the following CLI command to change the inspection mode to proxy:

```
config system settings  
    set inspection-mode proxy  
end
```

The **System Information** dashboard widget also shows the current **Mode**.

## How this guide is organized

This FortiOS Handbook chapter contains the following sections:

[Inside FortiOS: VoIP Protection](#) introduces FortiOS VoIP Protection

[FortiGate VoIP solutions: SIP](#) introduces FortiGate SIP support.

[Common SIP VoIP configurations](#) describes some common SIP configurations.

[SIP messages and media protocols](#) describes SIP messages and some common SIP media protocols.

[The SIP session helper](#) describes how the SIP session helper works and how to configure SIP support using the SIP session helper.

[The SIP ALG](#) describes how the SIP Application Layer Gateway (ALG) works and how to configure SIP support using the SIP ALG.

[Conflicts between the SIP ALG and the session helper](#) describes how to sort out conflicts between the SIP session helper and the ALG.



[Stateful SIP tracking, call termination, and session inactivity timeout](#) describes how the SIP ALG performs SIP stateful tracking, call termination and session activity timeouts.

[SIP and RTP/RTCP](#) describes how SIP relates to RTP and RTCP.

[How the SIP ALG creates RTP pinholes](#) describes how the SIP ALG creates pinholes.

[Configuration example: SIP in Transparent Mode](#) describes how to configure a FortiGate in Transparent mode to support SIP.

[RTP enable/disable \(RTP bypass\)](#) describes RTP bypass.

[Opening and closing SIP register, contact, via and record-route pinholes](#) describes how FortiOS opens and closes these pinholes.

[Accepting SIP register responses](#) describes how to enable accepting SIP register responses.

[How the SIP ALG performs NAT](#) describes how the SIP ALG performs NAT.

[Enhancing SIP pinhole security](#) describes how to open smaller pinholes.

[Hosted NAT traversal](#) describes SIP hosted NAT traversal and how to configure it.

[SIP over IPv6](#) describes how to configure SIP over IPv6.

[Deep SIP message inspection](#) describes how deep SIP message inspection works.

[Blocking SIP request messages](#) describes how to block SIP request messages to prevent some common SIP attacks.

[SIP rate limiting](#) includes more options for preventing SIP attacks.

[SIP logging](#) describes how to enable SIP logging.

[Inspecting SIP over SSL/TLS \(secure SIP\)](#) describes how to inspect encrypted SIP traffic.

[SIP and HA-session failover and geographic redundancy](#) describes how to use FGCP HA to support SIP geographic redundancy.

[SIP and IPS](#) describes how to turn on IPS for SIP sessions.

[SIP debugging](#) describes some tools for debugging your SIP configuration.

## FortiOS 5.6 VoIP new features

FortiOS 5.6 includes the following new VoIP and SIP features:

### SIP strict-register enabled by default in VoIP Profiles (380830)

If `strict-register` is disabled, when REGISTER is received by a FortiGate, the source address (usually the IP address of PBX) and ports (usually port 5060) are translated by NAT to the external address of the FortiGate and port 65476. Pinholes are then opened for SIP and RTP. This tells the SIP provider to send incoming SIP traffic to the external address of the FortiGate on port 65476.

This creates a security hole since the port is open regardless of the source IP address so an attacker who scans all the ports by sending REGISTER messages to the external IP of the FortiGate will eventually have one register go through.

When `strict-register` is enabled (the new default) the pinhole is smaller because it will only accept packets from the SIP server.

Enabling `strict-register` can cause problems when the SIP registrar and SIP proxy server are separate entities with separate IP addresses.

## SIP diagnose command improvements (376853)

A diagnose command has been added to the CLI that outputs VDOM data located in the voipd daemon.

```
diagnose sys sip-proxy vdom
```

### Example

```
(global) # diagnose sys sip-proxy vdom
VDOM list by id:
vdom 0 root (Kernel: root)
vdom 1 dmngmt-vdom (Kernel: dmngmt-vdom)
vdom 2 test2 (Kernel: test2)
vdom 3 test3 (Kernel: test3)
vdom 4 vdoma2 (Kernel: vdoma2)
vdom 5 vdomb2 (Kernel: vdomb2)
vdom 6 vdomc2 (Kernel: vdomc2)
vdom 7 vdoma (Kernel: vdoma)
vdom 8 vdomb (Kernel: vdomb)
vdom 9 vdomc (Kernel: vdomc)
VDOM list by name:
vdom 1 dmngmt-vdom (Kernel: dmngmt-vdom)
vdom 0 root (Kernel: root)
vdom 2 test2 (Kernel: test2)
vdom 3 test3 (Kernel: test3)
vdom 7 vdoma (Kernel: vdoma)
vdom 4 vdoma2 (Kernel: vdoma2)
vdom 8 vdomb (Kernel: vdomb)
vdom 5 vdomb2 (Kernel: vdomb2)
vdom 9 vdomc (Kernel: vdomc)
vdom 6 vdomc2 (Kernel: vdomc2)
```

# Inside FortiOS: Voice over IP (VoIP) Protection

The FortiOS SIP Application Layer Gateway (ALG) allows SIP calls to pass through a FortiGate by opening SIP and RTP pinholes and performing source and destination IP address and port translation for SIP and RTP packets.

There are a large number of SIP-related Internet Engineering Task Force (IETF) documents (Request for Comments) that define behavior of SIP and related applications. FortiOS completely support [RFC 3261](#) for SIP, [RFC 4566](#) for SDP and [RFC 3262](#) for Provisional Response Acknowledgment (PRACK). FortiOS also supports other SIP and SIP-related RFCs and performs Deep SIP message inspection for SIP statements defined in other SIP RFCs.

## Advanced Voice over IP Protection

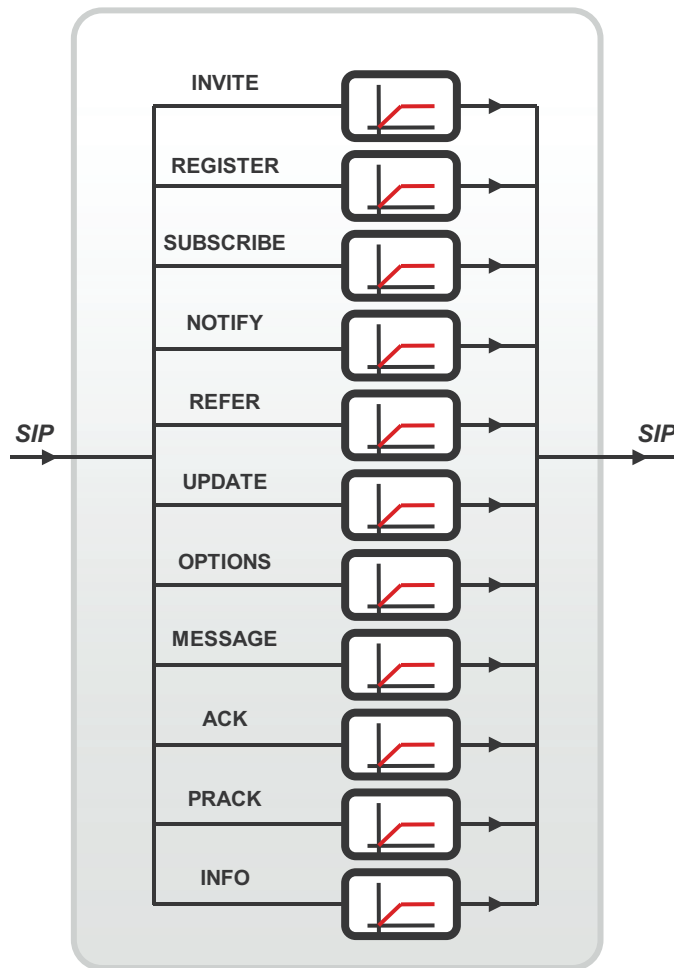
The FortiOS SIP Application Level Gateway (ALG) protects Voice over IP (SIP and SDP) services in Unified Communication and NGN/IMS networks with the following advanced VoIP defense mechanisms.

### Deep SIP message inspection (also called deep SIP header inspection)

Verifies SIP and SDP header syntax and protects SIP servers from potential SIP Fuzzing attacks. When a violation is detected, FortiOS can impose counter measures and can also send automatic SIP response messages to offload processing from the SIP server.

### SIP message rate limiting

Allows rate limiting of SIP messages per SIP request method. This prevents a SIP server from overload or from DoS attacks using particular SIP methods. For example, FortiOS can protect SIP servers from a flood of SIP REGISTER or INVITE messages, which can be caused by a DoS attack or a flash crowd.



## RTP and RTCP pinholing

RTP pinholing only forwards RTP/RTCP packets that conform to the particular session description of the associated SIP dialog. If a SIP dialog is finished, FortiOS automatically closes the pinhole. RTP/RTCP pinholing is supported by FortiASIC acceleration and achieves high packet throughput at low jitter and delay.

## Stateful SIP dialog tracking

FortiOS tracks SIP message sequences and prevents unwanted SIP messages that are not related to a particular SIP dialog. For instance, FortiOS can detect malicious SIP BYE messages that do not conform with the associated context of the SIP dialog.

## Inspecting SIP over SSL/TLS (secure SIP)

Some SIP phones and SIP servers use SSL or TLS to encrypt SIP signalling traffic. To allow SIP over SSL/TLS calls to pass through the FortiGate unit, the encrypted signalling traffic has to be unencrypted and inspected. FortiOS intercepts and unencrypts and inspects the SIP packets. Allowed packets are then re-encrypted and forwarded to their destination.

## Inspecting SIP on multiple ports

FortiOS can detect and inspect SIP and SDP UDP and TCP sessions and SIP SSL sessions and you can configure the ports that the SIP ALG monitors for these sessions. In addition you can configure two different ports for SIP UDP sessions and two different ports for SIP TCP sessions. The port configuration can be changed without affecting other parts of the SIP configuration.

## Carrier Grade protection

To protect VoIP infrastructure in carrier networks, FortiOS complies with typical carrier requirements for availability and robustness.

## High Availability

FortiOS supports a hot failover configuration with an active and a standby FortiGate device. FortiOS dynamically updates the context on the standby unit with SIP and RTP related data. This enables the standby unit to takeover stable voice calls in case of a planned or unplanned outage or failover of the active unit.

## Geographical Redundancy of SIP servers

In FortiOS SIP server cluster configurations the active and standby units can be deployed in different geographical locations. This configuration prevents a total outage of a SIP server infrastructure if one location goes offline. FortiOS supports the detection of SIP server outages (loss of heartbeats) and a redirect of SIP messages to the redundant SIP server location.

## Logging and Reporting

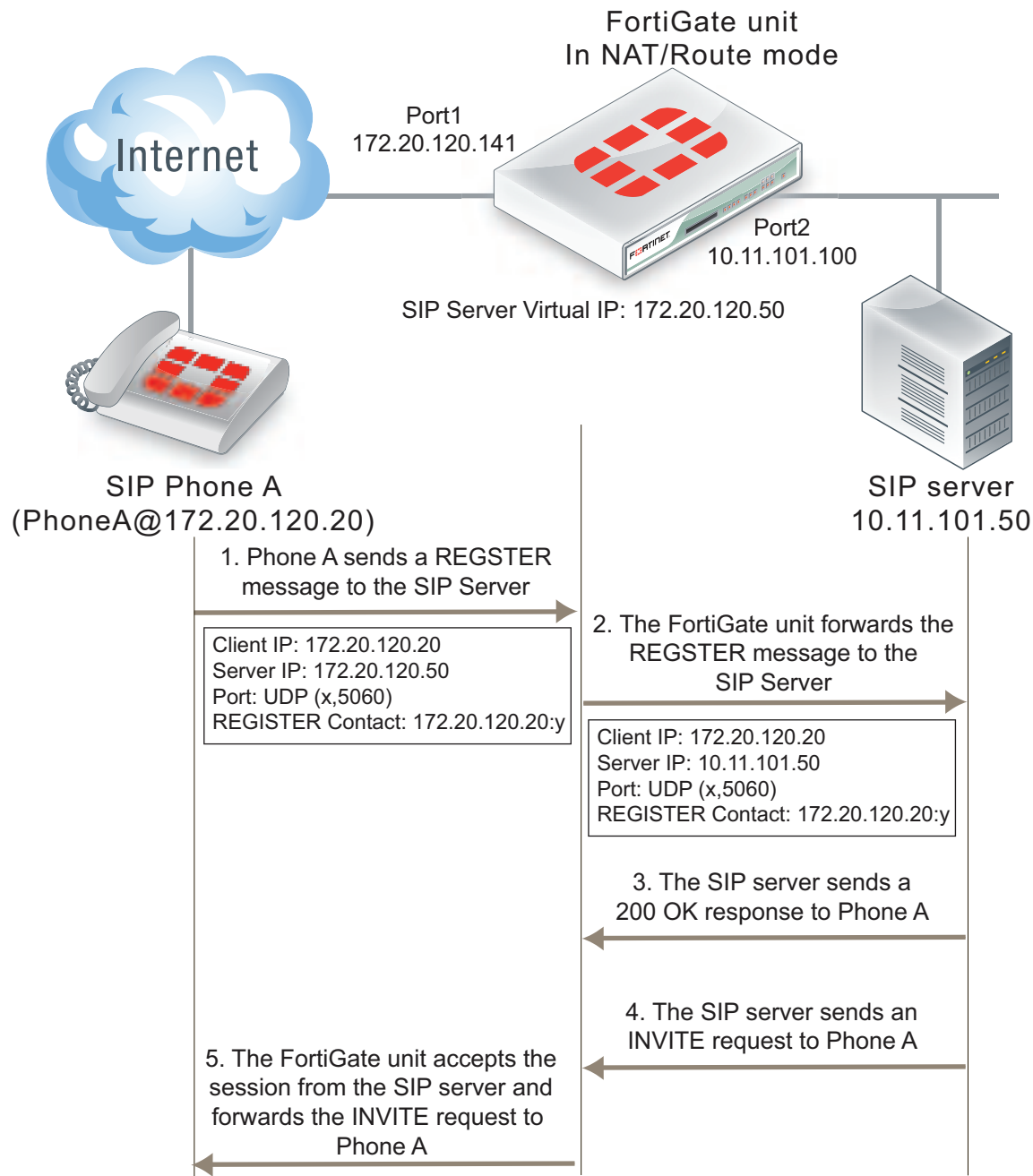
FortiOS can log call related information internally or to an external SYSLOG or FortiAnalyzer unit. This includes event logs that show particular SIP-related attacks or syntax violations with SIP messages or logs that summarize call statistics.

## NAT/NAPT

FortiOS performs configurable network address translation for IP addresses in the SIP and SDP header. The SIP ALG follows the configured NAT addresses in firewall virtual IPs and changes SIP header IP addresses accordingly. RTP NAT is controlled by SIP/SDP and the firewall policy. This allows translating an unlimited number of IP addresses without adding specific RTP policies.

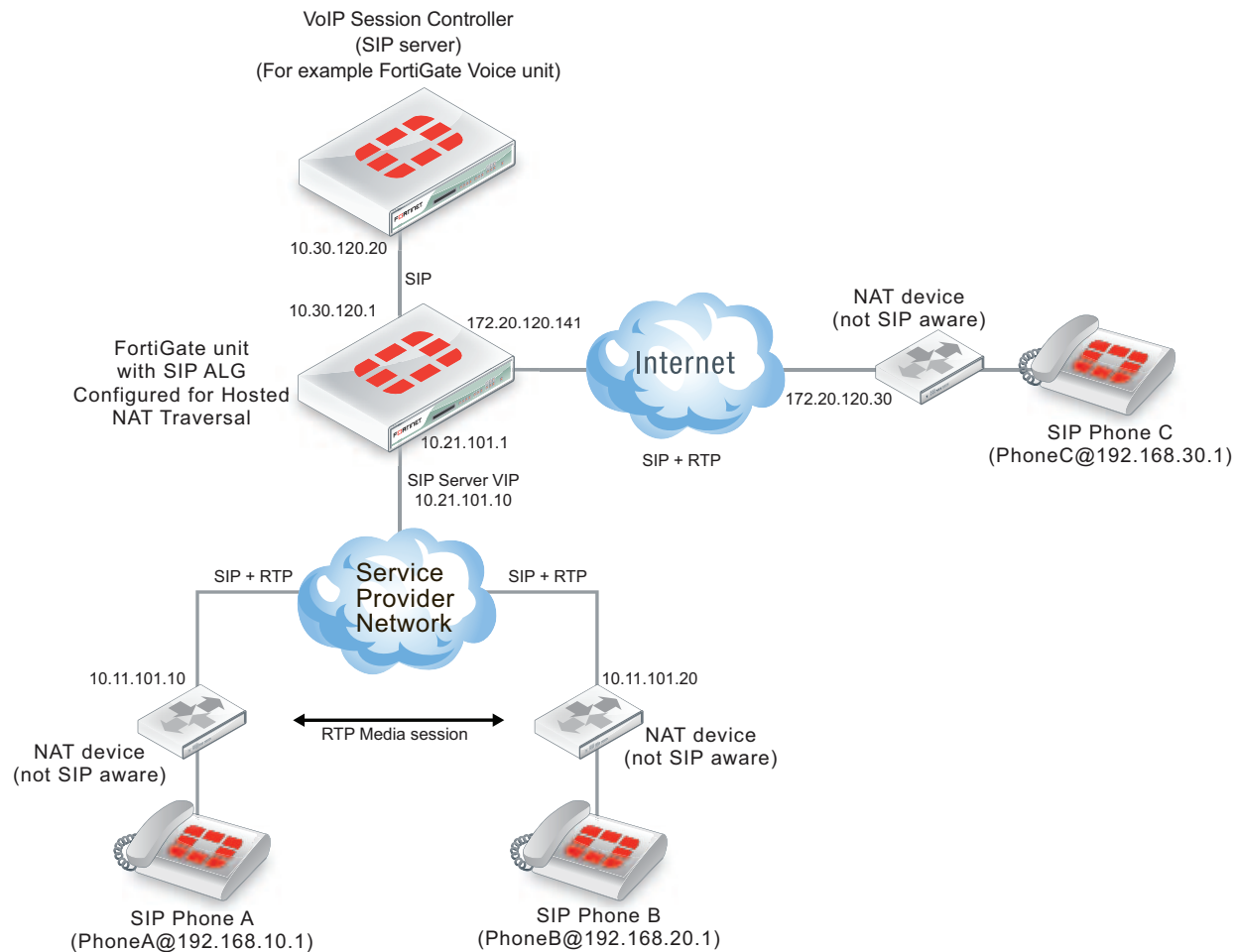
## Header Manipulation

FortiOS SIP and SDP header manipulation supports SIP Network Address Translation (NAT) through FortiGate units configured as NAT firewalls.



## Hosted NAT traversal (HNT)

In many service provider networks, CPE firewall devices provide NAT without application awareness. This causes issues for SIP/SDP and RTP traffic, since UAC IP address information references to the internal network behind the far end firewall. VoIP calls cannot be connected successfully. FortiOS mitigates far end NAT issues (called Hosted NAT traversal) by probing the first RTP packet from the UAC and learning the far end NA(P)T binding. FortiOS then updates the internal NAT binding for RTP accordingly.

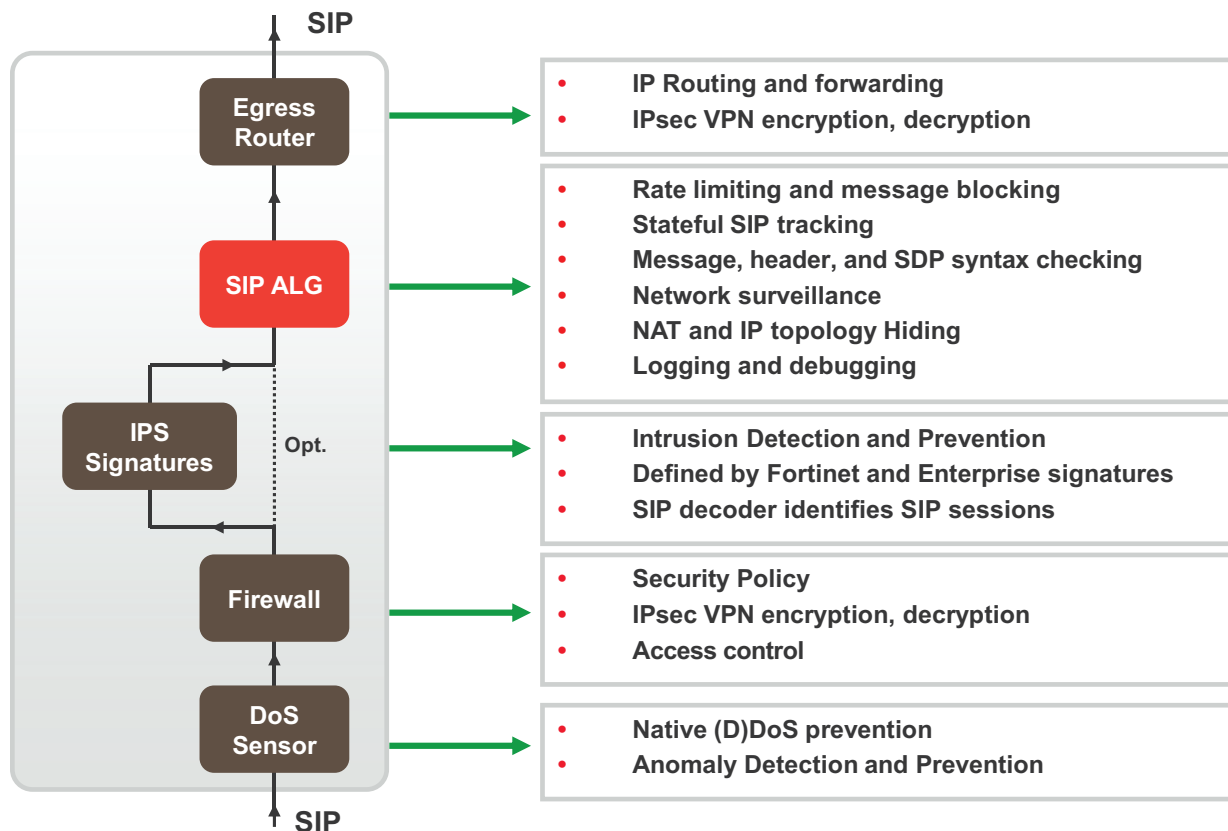


## IP Address conservation for NAT

In case of SIP and RTP NAT IP the original address information can get lost after translating to the provisioned IP addresses. This IP address information is sometimes required for detailed billing records or debugging purposes. FortiOS can maintain the original IP address information in a translated SIP header by adding it to the SIP/SDP info line (i=) or by adding it to the original attribute (o=). Either option can be selected depending on the SIP billing environment.

## SIP ALG activation

The FortiOS SIP ALG is applied to SIP traffic accepted by a firewall policy that includes a VoIP profile. The VoIP profile controls how the SIP ALG processes SIP sessions. FortiOS also includes a high-performance SIP session helper that provides limited SIP functionality. In most cases the SIP ALG should be used because the SIP ALG supports the complete range of FortiOS SIP features.



## SIP over IPv6

FortiOS, operating in NAT/Route and in Transparent mode supports SIP over IPv6. The SIP ALG can process SIP messages that use IPv6 addresses in the headers, bodies, and in the transport stack. The SIP ALG cannot modify the IPv6 addresses in the SIP headers so FortiGate units cannot perform SIP or RTP NAT over IPv6 and also cannot translate between IPv6 and IPv4 addresses.

## Platform support and hardware acceleration

FortiOS supports VoIP protection with the SIP ALG on all FortiGate hardware platforms. Whenever a FortiGate unit provides FortiASIC or SPM HW acceleration, the SIP ALG will use this option to fast-path RTP/RTCP traffic.

As well, since the SIP ALG is proxy-based, SIP control packets are not offloaded to NP4 or NP6 processors. But actual voice or other media traffic can be offloaded to NP4 or NP6 processors after the SIP session is established. Many FortiGate units also support low latency hardware acceleration configurations that also enhance SIP voice transmission.

FortiGate hardware acceleration provides a high throughput solution at very low jitter and delay. FortiOS provides efficient and highly scalable protection for VoIP in emerging Enterprise and Carrier network. This complements Fortinet's NGFW and UTM offerings. VoIP protection can be easily added to any firewall policy just by adding a VoIP profile.



VoIP protection is supported in FortiAnalyzer and FortiManager. Centralized logging and management are essential for carrier and MSSP service provider and are influencing business case calculations.

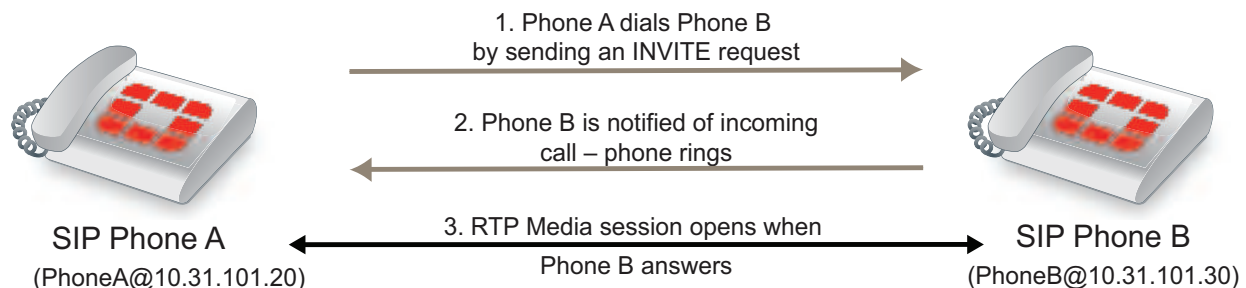
## Common SIP VoIP configurations

This section describes some common SIP VoIP configurations and simplified SIP dialogs for these configurations. This section also shows some examples of how adding a FortiGate affects SIP processing.

### Peer to peer configuration

In the peer to peer configuration shown below, two SIP phones (in the example, FortiFones) communicate directly with each other. The phones send SIP request and response messages back and forth between each other to establish the SIP session.

#### SIP peer to peer configuration

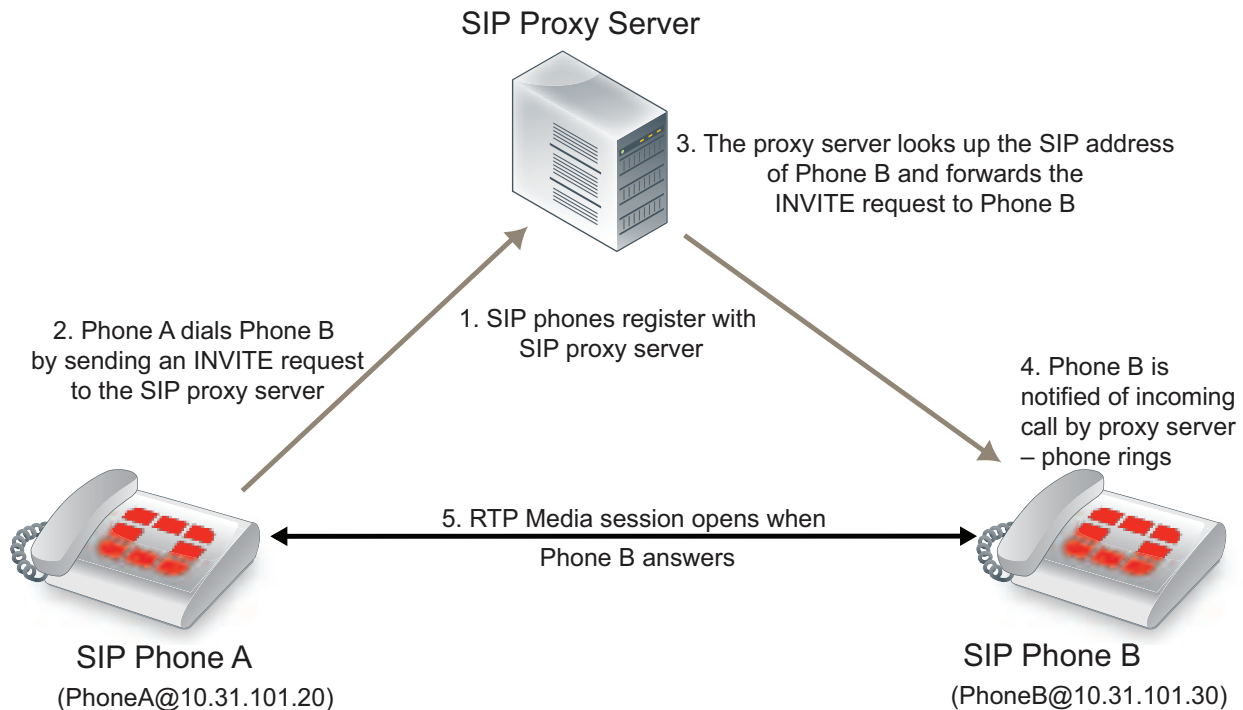


Peer to peer configurations are not very common because they require the SIP phones to keep track of the names and addresses of all of the other SIP phones that they can communicate with. In most cases a SIP proxy or re-direct server maintains addresses of a large number of SIP phones and a SIP phone starts a call by contacting the SIP proxy server.

### SIP proxy server configuration

A SIP proxy server acts as an intermediary between SIP phones and between SIP phones (for example, two FortiFones) and other SIP servers. As shown below, SIP phones send request and response messages to the SIP proxy server. The proxy server forwards the messages to other clients or to other SIP proxy servers. Proxy servers can hide SIP phones by proxying the signaling messages. To the other users on the VoIP network, the signaling invitations look as if they come from the SIP proxy server.

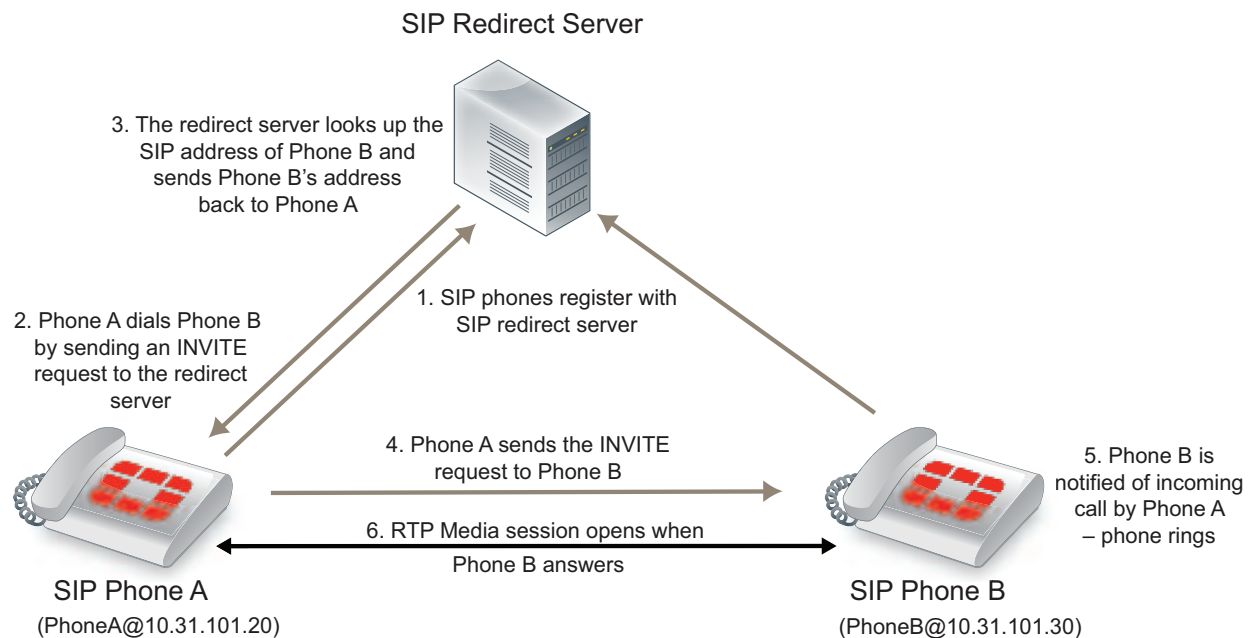
### SIP in proxy mode



A common SIP configuration would include multiple networks of SIP phones. Each of the networks would have its own SIP server. Each SIP server would proxy the communication between phones on its own network and between phones in different networks.

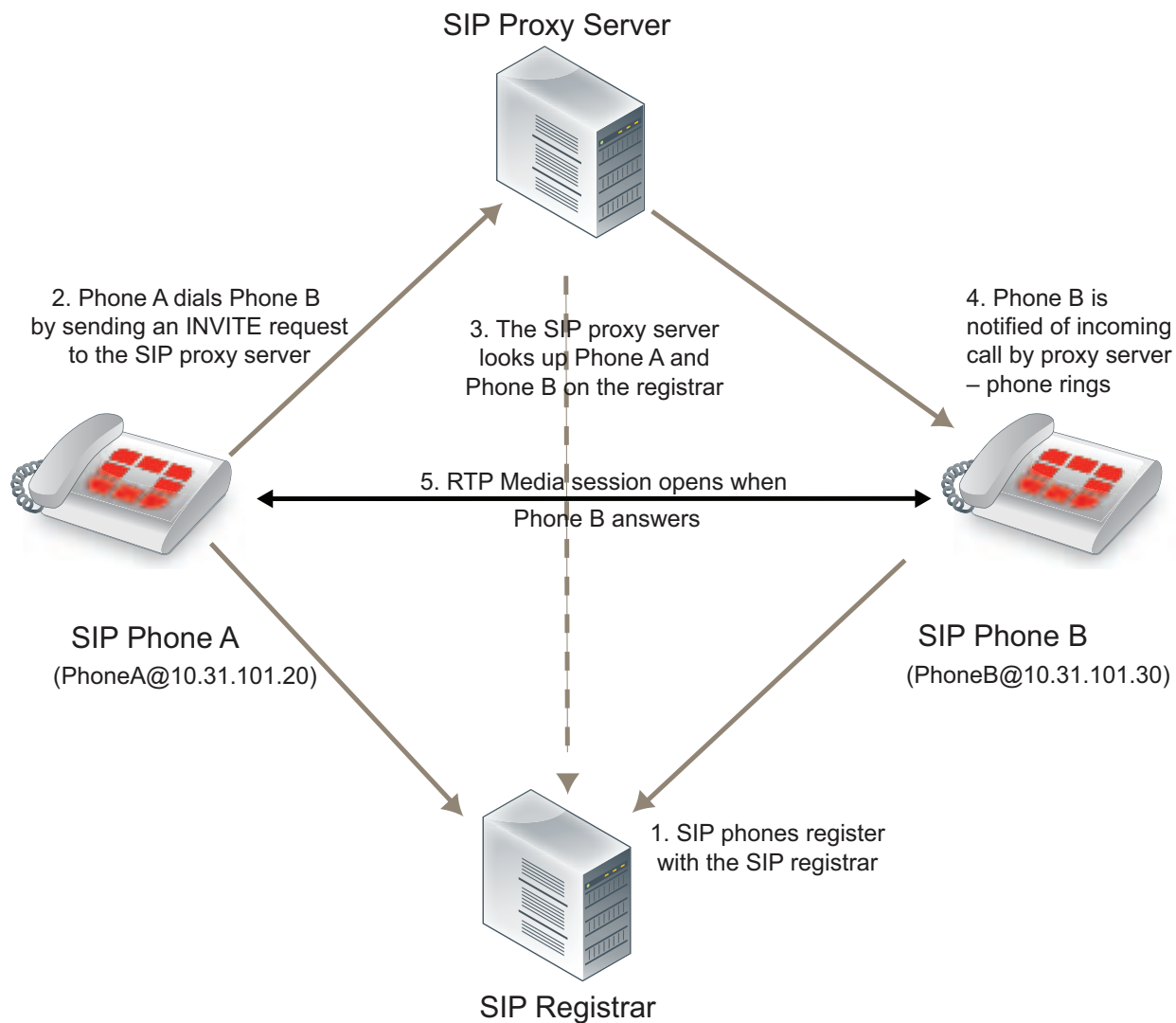
## SIP redirect server configuration

A SIP redirect server accepts SIP requests, maps the addresses in the request into zero or more new addresses and returns those addresses to the client. The redirect server does not initiate SIP requests or accept calls. As shown below, SIP clients send INVITE requests to the redirect server, which then looks up the destination address. The redirect server returns the destination address to the client. The client uses this address to send the INVITE request directly to the destination SIP client.

**SIP in redirect model****SIP registrar configuration**

A SIP registrar accepts SIP REGISTER requests from SIP phones for the purpose of updating a location database with this contact information. This database can then become a SIP location service that can be used by SIP proxy servers and redirect servers to locate SIP clients. As shown below, SIP clients send REGISTER requests to the SIP registrar.

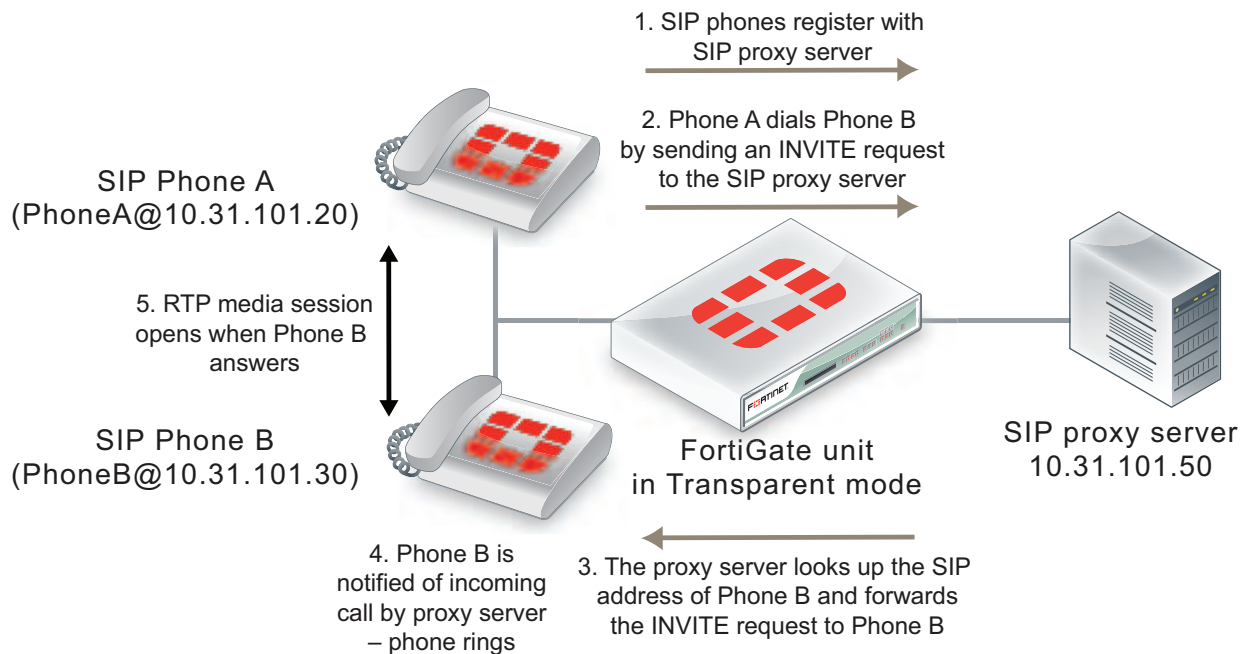
### SIP registrar and proxy servers



## SIP with a FortiGate

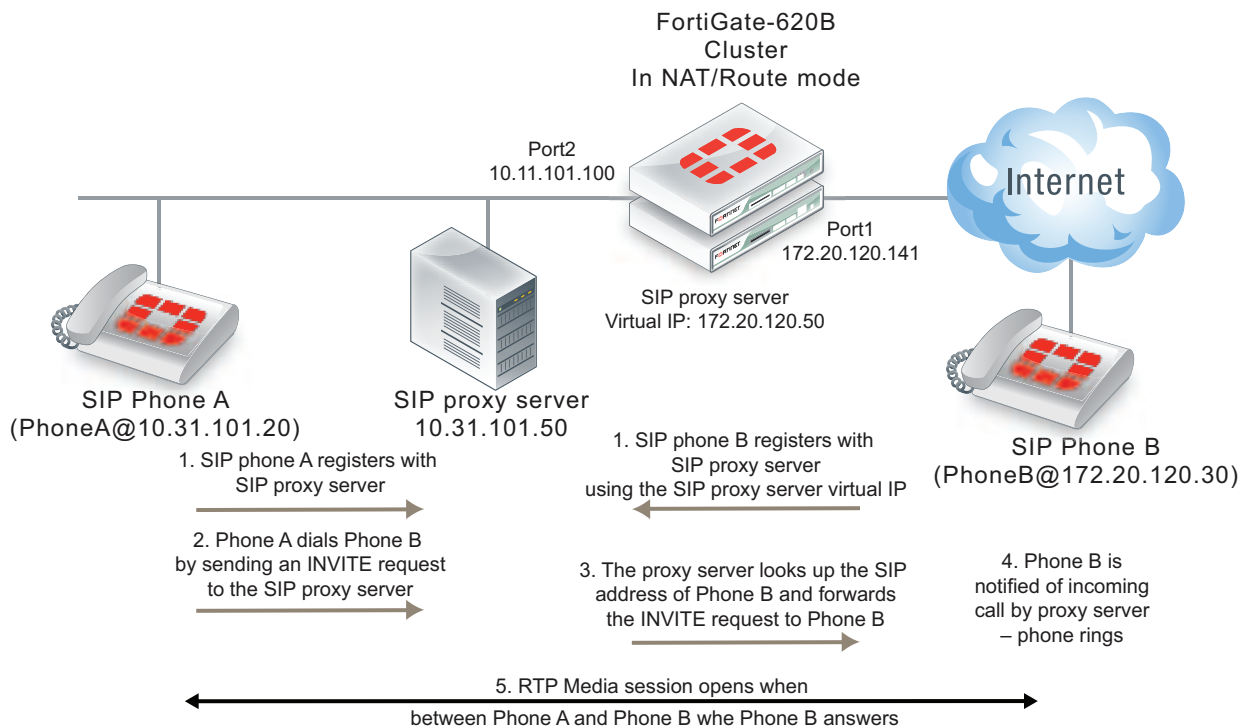
Depending on your security requirements and network configuration FortiGates may be in many different places in a SIP configuration. This section shows a few examples.

The diagram below shows a FortiGate installed between a SIP proxy server and SIP phones on the same network. The FortiGate is operating in Transparent mode so both the proxy server and the phones are on the same subnet. In this configuration, called SIP inspection without address translation, the FortiGate could be protecting the SIP proxy server on the private network by implementing SIP security features for SIP sessions between the SIP phones and the SIP proxy server.

**SIP network with FortiGate in Transparent mode**

The phones and server use the same SIP dialogs as they would if the FortiGate was not present. However, the FortiGate can be configured to control which devices on the network can connect to the SIP proxy server and can also protect the SIP proxy server from SIP vulnerabilities.

The following diagram shows a FortiGate operating in NAT/Route mode and installed between a private network and the Internet. Some SIP phones and the SIP proxy server are connected to the private network and some SIP phones are connected to the Internet. The SIP phones on the Internet can connect to the SIP proxy server through the FortiGate and communication between SIP phones on the private network and SIP phones on the Internet must pass through the FortiGate.

**SIP network with FortiGate in NAT/Route mode**

The phones and server use the same SIP dialog as they would if the FortiGate was not present. However, the FortiGate can be configured to control which devices on the network can connect to the SIP proxy server and can also protect the SIP proxy server from SIP vulnerabilities. In addition, the FortiGate has a firewall virtual IP that forwards packets sent to the SIP proxy server Internet IP address (172.20.120.50) to the SIP proxy server internal network IP address (10.31.101.30).

Since the FortiGate is operating in NAT/Route mode it must translate packet source and destination IP addresses (and optionally ports) as the sessions pass through the FortiGate. Also, the FortiGate must translate the addresses contained in the SIP headers and SDP body of the SIP messages. As well the FortiGate must open SIP and RTP pinholes through the FortiGate. SIP pinholes allow SIP signaling sessions to pass through the FortiGate between phones and between phones and SIP servers. RTP pinholes allow direct RTP communication between the SIP phones once the SIP dialog has established the SIP call. Pinholes are opened automatically by the FortiGate. Administrators do not add security policies for pinholes or for RTP sessions. All that is required is a security policy that accepts SIP traffic.

Opening an RTP pinhole means opening a port on a FortiGate interface to allow RTP traffic to use that port to pass through the FortiGate between the SIP phones on the Internet and SIP phones on the internal network. A pinhole only accepts packets from one RTP session. Since a SIP call involves at least two media streams (one from Phone A to Phone B and one from Phone B to Phone A) the FortiGate opens two RTP pinholes. Phone A sends RTP packets through a pinhole in port2 and Phone B sends RTP packets through a pinhole in port1. The FortiGate opens the pinholes when required by the SIP dialog and closes the pinholes when the SIP call is completed. The FortiGate opens new pinholes for each SIP call.

Each RTP pinhole actually includes two port numbers. The RTP port number as defined in the SIP message and an RTCP port number, which is the RTP port number plus 1. For example, if the SIP call used RTP port 3346 the FortiGate would create a pinhole for ports 3346 and 3347.

# SIP messages and media protocols

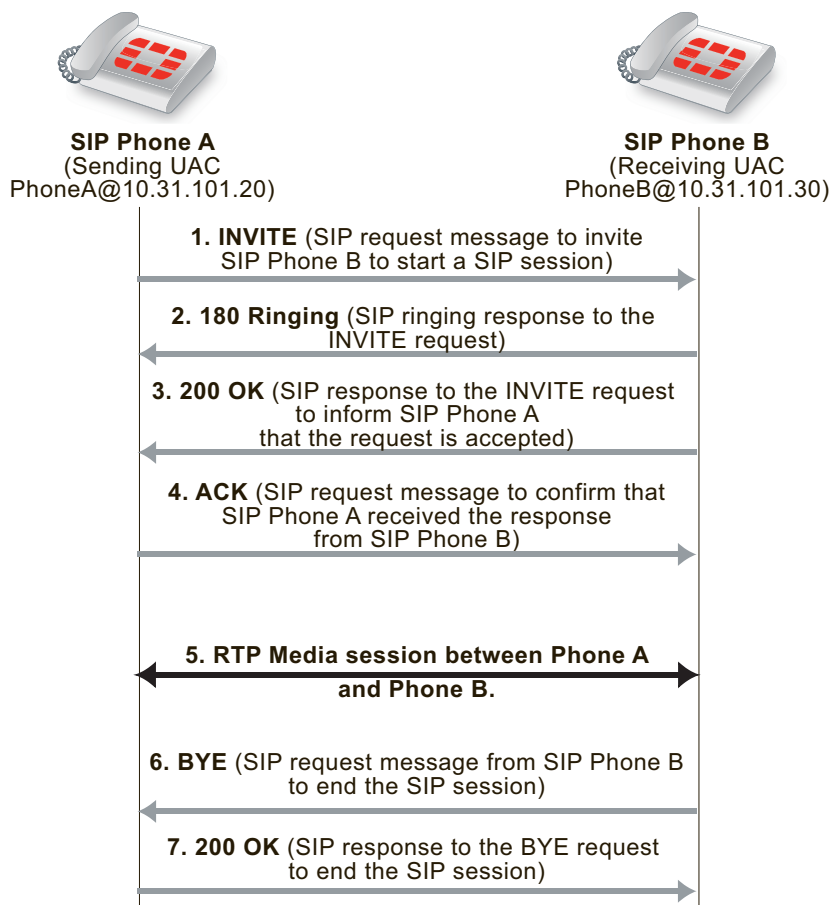
This section provides an overview of SIP messages and how they communicate information about SIP sessions and how SDP, RTP, and RTCP fits in with SIP communications.

SIP uses clear text messages to start, maintain, and end media sessions between SIP user agent clients (UACs) and user agent servers (UASs). These messages form a SIP dialog. A typical SIP dialog begins with an INVITE request message sent from a UAC to another UAC or to a UAS. The first INVITE request message attempts to start a SIP call and includes information about the sending UAC and the receiving UAC as well as information about the communication session.

If only two UACs are involved as shown below, the receiving UAC (Phone B) responds with a 180 Ringing and then a 200 OK SIP response message that informs Phone A that Phone B received and accepted the request. Phone A then sends an ACK message to notify Phone B that the SIP response was received. Phone A and Phone B can then participate in the RTP media session set up by the SIP messages.

When the phone call is complete, one of the UACs (in the example Phone B) hangs up sending a BYE request message to Phone A. Phone A then sends a 200 OK response to Phone B acknowledging that the session has ended.

## Basic SIP dialog between two UACs

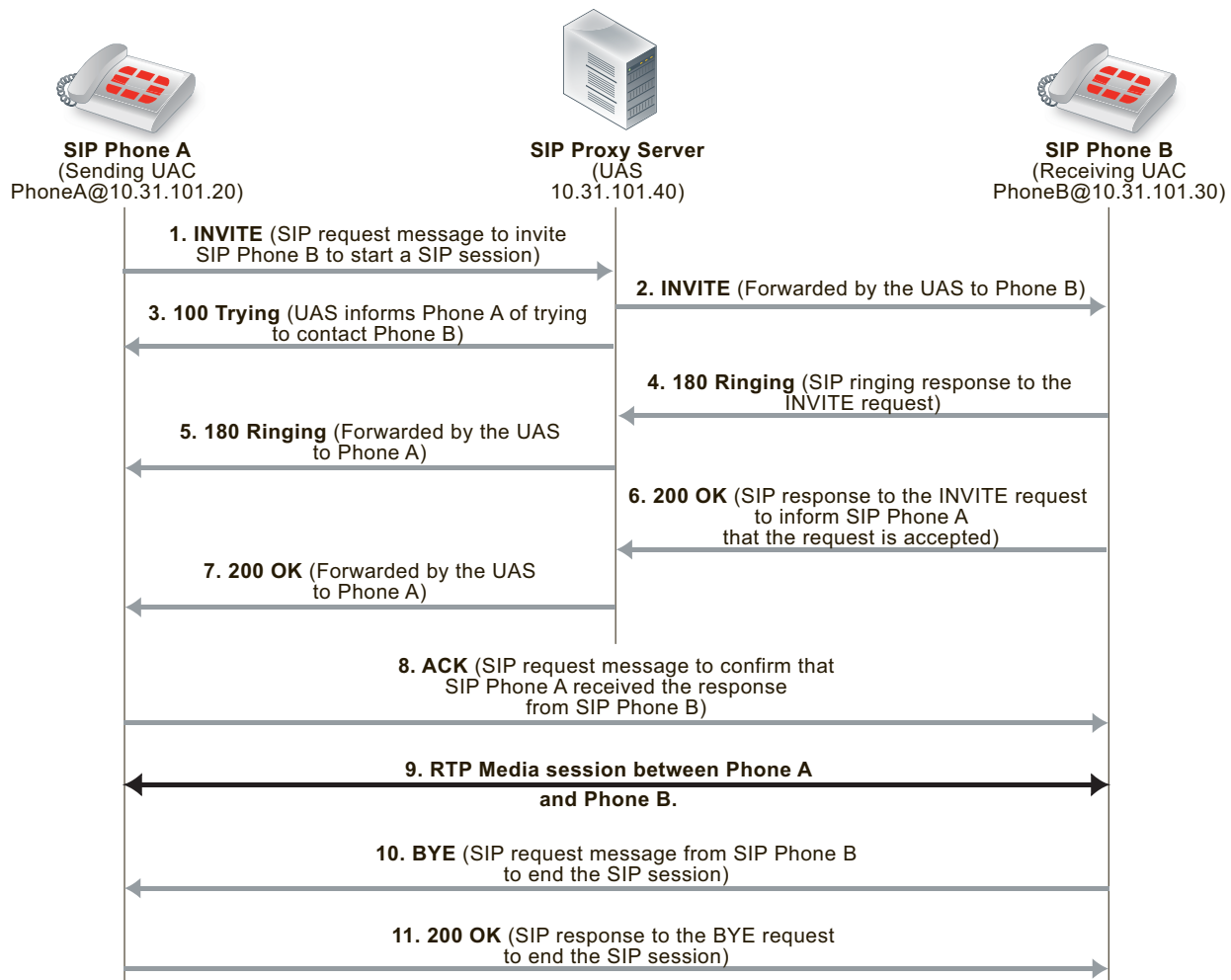




If a UAS in the form of a SIP proxy server is involved, similar messages are sent and received, but the proxy server participates as an intermediary in the initial call setup. In the example below the SIP proxy server receives the INVITE request from Phone A and forwards it to Phone B. The proxy server then sends a 100 Trying response to Phone A. Phone B receives the INVITE request and responds with a 180 Ringing and then a 200 OK SIP response message. These messages are received by the proxy server and forwarded to Phone A to notify Phone A that Phone B received and accepted the request. Phone A then sends an ACK message to notify Phone B that the SIP response was received. This response is received by the proxy server and forwarded to Phone B. Phone A and Phone B can then participate in the media session independently of the proxy server.

When the phone call is complete Phone B hangs up sending a BYE request message to Phone A. Phone A then sends a 200 OK response to Phone B acknowledging that the session has ended.

### Basic SIP dialog between UACs with a SIP proxy server UAS



The SIP messages include SIP headers that contain names and addresses of Phone A, Phone B and the proxy server. This addressing information is used by the UACs and the proxy server during the call set up.

The SIP message body includes Session Description Protocol (SDP) statements that Phone A and Phone B use to establish the media session. The SDP statements specify the type of media stream to use for the session (for example, audio for SIP phone calls) and the protocol to use for the media stream (usually the Real Time Protocol (RTP) media streaming protocol).

Phone A includes the media session settings that it would like to use for the session in the INVITE message. Phone B includes its response to these media settings in the 200 OK response. Phone A's ACK response confirms the settings that Phone A and Phone B then use for the media session.

## Hardware accelerated RTP processing

FortiGates can offload RTP packet processing to network processor (NP) interfaces. This acceleration greatly enhances the overall throughput and resulting in near speed RTP performance.

## SIP request messages

SIP sessions always start with a SIP request message (also just called a SIP request). SIP request messages also establish, maintain, and terminate SIP communication sessions. The following table lists some common SIP request message types.

### Common SIP request message types

| Message Type   | Description  |
|----------------|--|
| <b>INVITE</b>  | A client sends an INVITE request to invite another client to participate in a multimedia session. The INVITE request body usually contains the description of the session.   |
| <b>ACK</b>     | The originator of an INVITE message sends an ACK request to confirm that the final response to an INVITE request was received. If the INVITE request did not contain the session description, it must be included in the ACK request.  |
| <b>PRACK</b>   | In some cases, SIP uses provisional response messages to report on the progress of the response to a SIP request message. The provisional response messages are sent before the final SIP response message. Similar to an ACK request message, a PRACK request message is sent to acknowledge that a provisional response message has been received. |
| <b>OPTIONS</b> | The UA uses OPTIONS messages to get information about the capabilities of a SIP proxy. The SIP proxy server replies with a description of the SIP methods, session description protocols, and message encoding that are supported.   |
| <b>BYE</b>     | A client sends a BYE request to end a session. A BYE request from either end of the SIP session terminates the session.  |
| <b>CANCEL</b>  | A client sends a CANCEL request to cancel a previous INVITE request. A CANCEL request has no effect if the SIP server processing the INVITE sends a final response to the INVITE before receiving the CANCEL.  |

| Message Type  | Description   |
|---|---|
| <b>REGISTER</b>   | A client sends a REGISTER request to a SIP registrar server with information about the current location (IP address and so on) of the client. A SIP registrar server saves the information it receives in REGISTER requests and makes this information available to any SIP client or server attempting to locate the client. |
| <b>Info</b>   | For distributing mid-session signaling information along the signaling path for a SIP call. I   |
| <b>Subscribe</b>  | For requesting the current state and state updates of a remote node.  |
| <b>Notify</b>   | Informs clients and servers of changes in state in the SIP network.   |
| <b>Refer</b>  | Refers the recipient (identified by the Request-URI) to a third party according to the contact information in the request.  |
| <b>Update</b>   | Opens a pinhole for new or updated SDP information.   |
| <b>Response codes (1xx, 202, 2xx, 3xx, 4xx, 5xx, 6xx)</b> | Indicates the status of a transaction. For example: 200 OK, 202 Accepted, or 400 Bad Request.   |

## SIP response messages

SIP response messages (often just called SIP responses) provide status information in response to SIP request messages. All SIP response messages include a response code and a reason phrase. There are five SIP response message classes. They are described below.

There are also two types of SIP response messages, provisional and final. Final response messages convey the result of the request processing, and are sent reliably. Provisional responses provide information on the progress of the request processing, but may not be sent reliably. Provisional response messages start with 1xx and are also called informational response messages.

### Informational (or provisional)

Informational or provisional responses indicate that a request message was received and imply that the endpoint is going to process the request. Information messages may not be sent reliably and may not require an acknowledgement.

If the SIP implementation uses Provisional Response Acknowledgment (PRACK) ([RFC 3262](#)) then informational or provisional messages are sent reliably and require a PRACK message to acknowledge that they have been received.

Informational responses can contain the following reason codes and reason phrases:

```

100 Trying
180 Ringing
181 Call is being forwarded
182 Queued

```

183 Session progress

## Success

Success responses indicate that a request message was received, understood, and accepted. Success responses can contain the following reason codes and reason phrases:

200 OK  
202 Accepted

## Redirection

Redirection responses indicate that more information is required for the endpoint to respond to a request message. Redirection responses can contain the following reason codes and reason phrases:

300 Multiple choices  
301 Moved permanently  
302 Moved temporarily  
305 Use proxy  
380 Alternative service

## Client error

Client error responses indicate that a request message was received by a server that contains syntax that the server cannot understand (i.e. contains a syntax error) or cannot comply with. Client error responses include the following reason codes and reason phrases:

|   |                                   |
|---|-----------------------------------|
| 400 Bad request                         | 401 Unauthorized                  |
| 402 Payment required                    | 403 Forbidden                     |
| 404 Not found                           | 405 Method not allowed            |
| 406 Not acceptable                      | 407 Proxy authentication required |
| 408 Request time-out                    | 409 Conflict                      |
| 410 Gone                                | 411 Length required               |
| 413 Request entity too large            | 414 Request-URL too large         |
| 415 Unsupported media type              | 420 Bad extension                 |
| 480 Temporarily not available           |                                   |
| 481 Call leg/transaction does not exist |                                   |
| 482 Loop detected                       |                                   |
| 484 Address incomplete                  | 483 Too many hops                 |
| 486 Busy here                           | 485 Ambiguous                     |
| 488 Not acceptable here                 | 487 Request canceled              |

## Server error

Server error responses indicate that a server was unable to respond to a valid request message. Server error responses include the following reason codes and reason phrases:

500 Server internal error  
501 Not implemented  
502 Bad gateway  
502 Service unavailable  
504 Gateway time-out  
505 SIP version not supported

## Global failure

Global failure responses indicate that there are no servers available that can respond to a request message. Global failure responses include the following reason codes and reason phrases:

```
600 Busy everywhere
603 Decline
604 Does not exist anywhere
606 Not acceptable
```

## SIP message start line

The first line in a SIP message is called the start line. The start line in a request message is called the request-line and the start line in a response message is called the status-line.

|                     |   |
|---------------------|---|
| <b>Request-line</b> | <p>The first line of a SIP request message. The request-line includes the SIP message type, the SIP protocol version, and a Request URI that indicates the user or service to which this request is being addressed. The following example request-line specifies the INVITE message type, the address of the sender of the message (inviter@example.com), and the SIP version:</p> <pre>INVITE sip:inviter@example.com SIP/2.0</pre> |
| <b>Status-line</b>  | <p>The first line of a SIP response message. The status-line includes the SIP protocol version, the response code, and the reason phrase. The example status-line includes the SIP version, the response code (200) and the reason phrase (OK).</p> <pre>SIP/2.0 200 OK</pre>   |

## SIP headers

Following the start line, SIP messages contain SIP headers (also called SIP fields) that convey message attributes and to modify message meaning. SIP headers are similar to HTTP header fields and always have the following format:

```
<header_name>:<value>
```

SIP messages can include the SIP headers listed in the following table:

**SIP headers**

| SIP Header            | Description  |
|-----------------------|--|
| <b>Allow</b>          | <p>Lists the set of SIP methods supported by the UA generating the message. All methods, including ACK and CANCEL, understood by the UA MUST be included in the list of methods in the Allow header field, when present. For example:</p> <pre>Allow: INVITE, ACK, OPTIONS, CANCEL, BYE</pre>  |
| <b>Call-ID</b>        | <p>A globally unique identifier for the call, generated by the combination of a random string and the sender's host name or IP address. The combination of the To, From, and Call-ID headers completely defines a peer-to-peer SIP relationship between the sender and the receiver. This relationship is called a SIP dialog.</p> <pre>Call-ID: ddeg45e793@10.31.101.30</pre>   |
| <b>Contact</b>        | <p>Included in SIP request messages, the Contact header contains the SIP URI of the sender of the SIP request message. The receiver uses this URI to contact the sender. For example:</p> <pre>Contact: Sender &lt;sip:sender@10.31.100.20&gt;t</pre>  |
| <b>Content-Length</b> | <p>The number of bytes in the message body (in bytes).</p> <pre>Content-Length: 126</pre>  |
| <b>Content-Type</b>   | <p>In addition to SIP headers, SIP messages include a message body that contains information about the content or communication being managed by the SIP session. The Content-Type header specifies what the content of the SIP message is. For example, if you are using SIP with SDP, the content of the SIP message is SDP code.</p> <pre>Content-Type: application/sdp</pre> |
| <b>CSeq</b>           | <p>The command sequence header contains a sequence integer that is increased for each new SIP request message (but is not incremented in the response message). This header also includes the request name found in the request message request-line. For example:</p> <pre>CSeq: 1 INVITE</pre>   |
| <b>Expires</b>        | <p>Gives the relative time after which the message (or content) expires. The actual time and how the header is used depends on the SIP method. For example:</p> <pre>Expires: 5</pre>  |
| <b>From</b>           | <p>Identifies the sender of the message. Responses to a message are sent to the address of the sender. The following example includes the sender's name (Sender) and the sender's SIP address (sender@10.31.101.20.):</p> <pre>From: Sender &lt;sip:sender@10.31.101.20&gt;</pre>  |

| SIP Header                 | Description  |
|----------------------------|--|
| <b>Max-forwards</b>        | <p>An integer in the range 0-255 that limits the number of proxies or gateways that can forward the request message to the next downstream server. Also called the number of hops, this value is decreased every time the message is forwarded. This can also be useful when the client is attempting to trace a request chain that appears to be failing or looping in mid-chain.</p> <p>For example: <code>Max-Forwards: 30</code></p> |
| <b>P-Asserted-Identity</b> | <p>The P-Asserted-Identity header is used among trusted SIP entities to carry the identity of the user sending a SIP message as it was verified by authentication. See <a href="#">RFC 3325</a>. The header contains a SIP URI and an optional display-name, for example:</p> <p><code>P-Asserted-Identity: "Example Person" &lt;sip:10.31.101.50&gt;</code></p>   |
| <b>RAck</b>                | <p>Sent in a PRACK request to support reliability of information or provisional response messages. It contains two numbers and a method tag. For example:</p> <p><code>RAck: 776656 1 INVITE</code></p>  |
| <b>Record-Route</b>        | <p>Inserted into request messages by a SIP proxy to force future requests to be routed through the proxy. In the following example, the host at IP address 10.31.101.50 is a SIP proxy. The <code>lr</code> parameter indicates the URI of a SIP proxy in Record-Route headers.</p> <p><code>Record-Route: &lt;sip:10.31.101.50;lr&gt;</code></p>  |
| <b>Route</b>               | <p>Forces routing for a request message through one or more SIP proxies. The following example includes two SIP proxies:</p> <p><code>Route: &lt;sip:172.20.120.10;lr&gt;, &lt;sip:10.31.101.50;lr&gt;</code></p>  |
| <b>RSeq</b>                | <p>The RSeq header is used in information or provisional response messages to support reliability of informational response messages. The header contains a single numeric value. For example:</p> <p><code>RSeq: 33456</code></p>   |
| <b>To</b>                  | <p>Identifies the receiver of the message. The address in this field is used to send the message to the receiver. The following example includes the receiver's name (<code>Receiver</code>) and the receiver's SIP address (<code>receiver@10.31.101.30</code>):</p> <p><code>To: Receiver &lt;sip:receiver@10.31.101.30&gt;</code></p>   |
| <b>Via</b>                 | <p>Indicates the SIP version and protocol to be used for the SIP session and the address to which to send the response to the message that contains the Via field. The following example Via field indicates to use SIP version 2, UDP for media communications, and to send the response to 10.31.101.20 using port 5060.</p> <p><code>Via: SIP/2.0/UDP 10.31.101.20:5060</code></p>  |

## The SIP message body and SDP session profiles

The SIP message body describes the session to be initiated. For example, in a SIP phone call the body usually includes audio codec types, sampling rates, server IP addresses and so on. For other types of SIP session the body could contain text or binary data of any type which relates in some way to the session. The message body is included in request and response messages.

Two possible SIP message body types:

- Session Description Protocol (SDP), most commonly used for SIP VoIP.
- Multipurpose Internet Mail Extensions (MIME)

SDP is most often used for VoIP and FortiGate supports SDP content in SIP message bodies. SDP is a text-based protocol used by SIP to control media sessions. SDP does not deliver media but provides a session profile that contains media details, transport addresses, parameter negotiation, and other session description metadata for the participants in a media session. The participants use the information in the session profile to negotiate how to communicate and to manage the media session. SDP is described by [RFC 4566](#).

An SDP session profile always contains session information and may contain media information. Session information appears at the start of the session profile and media information (using the `m=` attribute) follows.

SDP session profiles can include the attributes listed in the following table.

### SDP session profile attributes

| Attribute | Description   |
|-----------|---|
| <b>a=</b> | Attributes to extend SDP in the form <code>a=&lt;attribute&gt;</code> or <code>a=&lt;attribute&gt;:&lt;value&gt;</code> .   |
| <b>b=</b> | Contains information about the bandwidth required for the session or media in the form <code>b=&lt;bandwidth_type&gt;:&lt;bandwidth&gt;</code> .  |
| <b>c=</b> | Connection data about the session including the network type (usually IN for Internet), address type (IPv4 or IPv6), the connection source address, and other optional information. For example:<br><br><code>c=IN IPv4 10.31.101.20</code> |
| <b>i=</b> | A text string that contains information about the session. For example:<br><br><code>i=A audio presentation about SIP</code>  |
| <b>k=</b> | Can be used to convey encryption keys over a secure and trusted channel. For example:<br><br><code>k=clear:444gdduudjffdee</code>   |



| Attribute | Description   |
|-----------|---|
| <b>m=</b> | <p>Media information, consisting of one or more lines all starting with <b>m=</b> and containing details about the media including the media type, the destination port or ports used by the media, the protocol used by the media, and a media format description.</p> <pre>m=audio 49170 RTP 0 3 m-video 3345/2 udp 34 m-video 2910/2 RTP/AVP 3 56</pre> <p>Multiple media lines are needed if SIP is managing multiple types of media in one session (for example, separate audio and video streams).</p> <p>Multiple ports for a media stream are indicated using a slash. <code>3345/2 udp</code> means UDP ports 3345 and 3346. Usually RTP uses even-numbered ports for data with the corresponding one-higher odd ports used for the RTCP session belonging to the RTP session. So <code>2910/2 RTP/AVP</code> means ports 2910 and 2912 are used for RTP and 2911 and 2913 are used for RTCP.</p> <p>Media types include <code>udp</code> for an unspecified protocol that uses UDP, <code>RTP</code> or <code>RTP/AVP</code> for standard RTP and <code>RTP/SAVP</code> for secure RTP.</p> |
| <b>o=</b> | <p>The sender's username, a session identifier, a session version number, the network type (usually IN for Internet), the address type (for example, IPv4 or IPv6), and the sending device's IP address. The <b>o=</b> field becomes a universal identifier for this version of this session description. For example:</p> <pre>o=PhoneA 5462346 332134 IN IP4 10.31.101.20</pre>   |
| <b>r=</b> | <p>Repeat times for a session. Used if a session will be repeated at one or more timed intervals. Not normally used for VoIP calls. The times can be in different formats. For example:</p> <pre>r=7d 1h 0 25h r=604800 3600 0 90000</pre>  |
| <b>s=</b> | <p>Any text that describes the session or <b>s=</b> followed by a space. For example:</p> <pre>s=Call from inviter</pre>  |
| <b>t=</b> | <p>The start and stop time of the session. Sessions with no time restrictions (most VoIP calls) have a start and stop time of 0.</p> <pre>t=0 0</pre>   |
| <b>v=</b> | <p>SDP protocol version. The current SDP version is 0 so the <b>v=</b> field is always:</p> <pre>v=0</pre>  |
| <b>z=</b> | <p>Time zone adjustments. Used for scheduling repeated sessions that span the time between changing from standard to daylight savings time.</p> <pre>z=2882844526 -1h 2898848070 0</pre>  |

## Example SIP messages

The following example SIP INVITE request message was sent by PhoneA to PhoneB. The first nine lines are the SIP headers. The SDP profile starts with v=0 and the media part of the session profile is the last line, starting with m=.

```
INVITE sip:PhoneB@172.20.120.30 SIP/2.0
Via: SIP/2.0/UDP 10.31.101.50:5060
From: PhoneA <sip:PhoneA@10.31.101.20>
To: PhoneB <sip:PhoneB@172.20.120.30>
Call-ID: 314159@10.31.101.20
CSeq: 1 INVITE
Contact: sip:PhoneA@10.31.101.20
Content-Type: application/sdp
Content-Length: 124
v=0
o=PhoneA 5462346 332134 IN IP4 10.31.101.20
s=Let's Talk
t=0 0
c=IN IP4 10.31.101.20
m=audio 49170 RTP 0 3
```

The following example shows a possible 200 OK SIP response message in response to the previous INVITE request message. The response includes 200 OK which indicates success, followed by an echo of the original SIP INVITE request followed by PhoneB's SDP profile.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.31.101.50:5060
From: PhoneA <sip:PhoneA@10.31.101.20>
To: PhoneB <sip:PhoneB@172.20.120.30>
Call-ID: 314159@10.31.101.20
CSeq: 1 INVITE
Contact: sip:PhoneB@10.31.101.30
Content-Type: application/sdp
Content-Length: 107
v=0
o=PhoneB 124333 67895 IN IP4 172.20.120.30
s=Hello!
t=0 0
c=IN IP4 172.20.120.30
m=audio 3456 RTP 0
```

SIP can support multiple media streams for a single SIP session. Each media stream will have its own c= and m= lines in the body of the message. For example, the following message includes three media streams:

```
INVITE sip:PhoneB@172.20.120.30 SIP/2.0
Via: SIP/2.0/UDP 10.31.101.20:5060
From: PhoneA <sip:PhoneA@10.31.101.20>
To: PhoneB <sip:PhoneB@172.20.120.30>
Call-ID: 314159@10.31.101.20
CSeq: 1 INVITE
Contact: sip:PhoneA@10.31.101.20
Content-Type: application/sdp
Content-Length: 124
v=0
o=PhoneA 5462346 332134 IN IP4 10.31.101.20
s=Let's Talk
```

```
t=0 0
c=IN IP4 10.31.101.20
m=audio 49170 RTP 0 3
c=IN IP4 10.31.101.20
m=audio 49172 RTP 0 3
c=IN IP4 10.31.101.20
m=audio 49174 RTP 0 3
```

# The SIP session helper

The SIP session-helper is a high-performance solution that provides basic support for SIP calls passing through the FortiGate by opening SIP and RTP pinholes and by performing NAT of the addresses in SIP messages.

The SIP session helper:

- Understands SIP dialog messages.
- Keeps the states of the SIP transactions between SIP UAs and SIP servers.
- Translates SIP header and SDP information to account for NAT operations performed by the FortiGate.
- Opens up and closes dynamic SIP pinholes for SIP signalling traffic.
- Opens up and closes dynamic RTP and RTSP pinholes for RTP and RTSP media traffic.
- Provides basic SIP security as an access control device.
- Uses the intrusion protection (IPS) engine to perform basic SIP protocol checks.

## SIP session helper configuration overview

By default FortiOS uses the SIP ALG for SIP traffic. If you want to use the SIP session helper you need to enter the following command:

```
config system settings
    set default-voip-alg-mode kernel-helper-based
end
```

The SIP session helper is set to listen for SIP traffic on TCP or UDP port 5060. SIP sessions using port 5060 accepted by a security policy that does not include a VoIP profile are processed by the SIP session helper.

You can enable and disable the SIP session helper, change the TCP or UDP port that the session helper listens on for SIP traffic, and enable or disable SIP NAT tracing. If the FortiGate is operating with multiple VDOMs, each VDOM can have a different SIP session helper configuration.

To have the SIP session helper process SIP sessions you need to add a security policy that accepts SIP sessions on the configured SIP UDP or TCP ports. The security policies can have service set to ANY, or to the SIP pre-defined firewall service, or a custom firewall service. The SIP pre-defined firewall service restricts the security policy to only accepting sessions on UDP port 5060.

If NAT is enabled for security policies that accept SIP traffic, the SIP session helper translates addresses in SIP headers and in the RDP profile and opens up pinholes as required for the SIP traffic. This includes security policies that perform source NAT and security policies that contain virtual IPs that perform destination NAT and port forwarding. No special SIP configuration is required for this address translation to occur, it is all handled automatically by the SIP session helper according to the NAT configuration of the security policy that accepts the SIP session.

To use the SIP session helper you must not add a VoIP profile to the security policy. If you add a VoIP profile, SIP traffic bypasses the SIP session helper and is processed by the SIP ALG.



In most cases you would want to use the SIP ALG since the SIP session helper provides limited functionality. However, the SIP session helper is available and can be useful for high-performance solutions where a high level of SIP security is not a requirement.

## Disabling and enabling the SIP session helper

You can use the following steps to disable the SIP session helper. You might want to disable the SIP session helper if you don't want the FortiGate to apply NAT or other SIP session help features to SIP traffic. With the SIP session helper disabled, the FortiGate can still accept SIP sessions if they are allowed by a security policy, but the FortiGate will not be able to open pinholes or NAT the addresses in the SIP messages.

### To disable the sip session helper

1. Enter the following command to find the sip session helper entry in the session-helper list:

```
show system session-helper
.
.
.
edit 13
    set name sip
    set port 5060
    set protocol 17
next
.
.
.
```

This command output shows that the sip session helper listens in UDP port 5060 for SIP sessions.

2. Enter the following command to delete session-helper list entry number 13 to disable the sip session helper:

```
config system session-helper
    delete 13
end
```

If you want to use the SIP session helper you can verify whether it is enabled or disabled using the `show system session-helper` command.



You do not have to disable the SIP session helper to use the SIP ALG.

If the SIP session helper has been disabled by being removed from the session-helper list you can use the following command to enable the SIP session helper by adding it back to the session helper list:

```
config system session-helper
    edit 0
        set name sip
        set port 5060
        set protocol 17
    end
```

## Changing the port numbers that the SIP session helper listens on

You can use the following command to change the port number that the SIP session helper listens on for SIP traffic to 5064. The SIP session helper listens on the same port number for UDP and TCP SIP sessions. In this example, the SIP session helper is session helper 13:

```
config system session-helper
  edit 13
    set port 5064
  end
```



The `config system settings options sip-tcp-port, sip-udp-port, and sip-ssl-port` control the ports that the SIP ALG listens on for SIP sessions. See [Changing the port numbers that the SIP ALG listens on on page 47](#).

Your FortiGate may use a different session helper number for SIP. Enter the following command to view the session helpers:

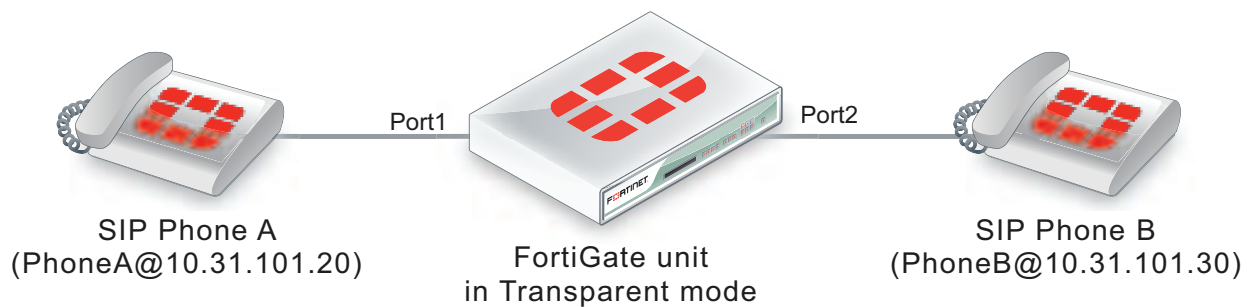
```
show system session-helper
.
.
.
edit 13
  set name sip
  set port 5060
  set protocol 17
end
.
.
.
```

## Configuration example: SIP session helper in Transparent Mode

The figure below shows an example SIP network consisting of a FortiGate operating in Transparent mode between two SIP phones. Since the FortiGate is operating in Transparent mode both phones are on the same network and the FortiGate and the SIP session helper does not perform NAT. Even though the SIP session helper is not performing NAT you can use this configuration to apply SIP session helper security features to the SIP traffic.

The FortiGate requires two security policies that accept SIP packets. One to allow SIP Phone A to start a session with SIP Phone B and one to allow SIP Phone B to start a session with SIP Phone A.

### SIP network with FortiGate in Transparent mode



## General configuration steps

The following general configuration steps are required for this SIP configuration that uses the SIP session helper. This example includes security policies that specifically allow SIP sessions using UDP port 5060 from Phone A to Phone B and from Phone B to Phone A. In most cases you would have more than two phones so would use more general security policies. Also, you can set the firewall service to ANY to allow traffic other than SIP on UDP port 5060.

This example assumes that you have entered the following command to enable using the SIP session helper:

```
config system settings
    set default-voip-alg-mode kernel-helper-based
end
```

1. Add firewall addresses for Phone A and Phone B.
2. Add a security policy that accepts SIP sessions initiated by Phone A.
3. Add a security policy that accepts SIP sessions initiated by Phone B.

## Configuration steps - GUI

### To add firewall addresses for the SIP phones

1. Go to **Policy & Objects > Addresses**.
2. Select **Create New > Address** to add the following addresses for Phone A and Phone B:

| Category          | Address                      |
|-------------------|------------------------------|
| Name              | Phone_A                      |
| Type              | IP/Netmask                   |
| Subnet / IP Range | 10.31.101.20/255.255.255.255 |
| Interface         | port1                        |

| Category | Address |
|----------|---------|
|----------|---------|

|                          |                              |
|--------------------------|------------------------------|
| <b>Name</b>              | Phone_B                      |
| <b>Type</b>              | IP/Netmask                   |
| <b>Subnet / IP Range</b> | 10.31.101.30/255.255.255.255 |
| <b>Interface</b>         | port2                        |

#### To add security policies to accept SIP sessions

1. Go to **Policy & Objects > IPv4 Policy**.
2. Select **Create New** to add a security policy.
3. Add a security policy to allow Phone A to send SIP request messages to Phone B:

|                            |         |
|----------------------------|---------|
| <b>Incoming Interface</b>  | port1   |
| <b>Outgoing Interface</b>  | port2   |
| <b>Source</b>              | Phone_A |
| <b>Destination Address</b> | Phone_B |
| <b>Schedule</b>            | always  |
| <b>Service</b>             | SIP     |
| <b>Action</b>              | ACCEPT  |

4. Select **OK**.
5. Add a security policy to allow Phone B to send SIP request messages to Phone A:

|                            |         |
|----------------------------|---------|
| <b>Incoming Interface</b>  | port2   |
| <b>Outgoing Interface</b>  | port1   |
| <b>Source Address</b>      | Phone_B |
| <b>Destination Address</b> | Phone_A |
| <b>Schedule</b>            | always  |
| <b>Service</b>             | SIP     |
| <b>Action</b>              | ACCEPT  |

6. Select **OK**.

## Configuration steps - CLI

#### To add firewall addresses for Phone A and Phone B and security policies to accept SIP sessions

1. Enter the following command to add firewall addresses for Phone A and Phone B.



```

config firewall address
edit Phone_A
    set associated interface port1
    set type ipmask
    set subnet 10.31.101.20 255.255.255.255
next
edit Phone_B
    set associated interface port2
    set type ipmask
    set subnet 10.31.101.30 255.255.255.255
end

```

2. Enter the following command to add security policies to allow Phone A to send SIP request messages to Phone B and Phone B to send SIP request messages to Phone A.

```

config firewall policy
edit 0
    set srcintf port1
    set dstintf port2
    set srcaddr Phone_A
    set dstaddr Phone_B
    set action accept
    set schedule always
    set service SIP
next
edit 0
    set srcintf port2
    set dstintf port1
    set srcaddr Phone_B
    set dstaddr Phone_A
    set action accept
    set schedule always
    set service SIP
    set utm-status enable
end

```

## SIP session helper diagnose commands

You can use the `diagnose sys sip` commands to display diagnostic information for the SIP session helper.

Use the following command to set the debug level for the SIP session helper. Different debug masks display different levels of detail about SIP session helper activity.

```
diagnose sys sip debug-mask <debug_mask_int>
```

Use the following command to display the current list of SIP dialogs being processed by the SIP session helper. You can also use the `clear` option to delete all active SIP dialogs being processed by the SIP session helper.

```
diagnose sys sip dialog {clear | list}
```

Use the following command to display the current list of SIP NAT address mapping tables being used by the SIP session helper.

```
diagnose sys sip mapping list
```

Use the following command to display the current SIP session helper activity including information about the SIP dialogs, mappings, and other SIP session help counts. This command can be useful to get an overview of what the SIP session helper is currently doing.

```
diagnose sys sip status
```

## The SIP ALG

In most cases you should use the SIP Application Layer Gateway (ALG) for processing SIP sessions. The SIP ALG provides the same basic SIP support as the SIP session helper. Additionally, the SIP ALG provides a wide range of features that protect your network from SIP attacks, apply rate limiting to SIP sessions, check the syntax of SIP and SDP content of SIP messages, and provide detailed logging and reporting of SIP activity.

By default all SIP traffic is processed by the SIP ALG. If the policy that accepts the SIP traffic includes a VoIP profile the SIP traffic is processed by that profile. If the policy does not include a SIP profile the SIP traffic is processed by the SIP ALG using the default VoIP profile.

If a FortiGate or a VDOM has been configured to use the SIP session helper, you can change this behavior to the default configuration of using the SIP ALG with the following command:

```
config system settings
    set default-voip-alg-mode proxy-based
end
```



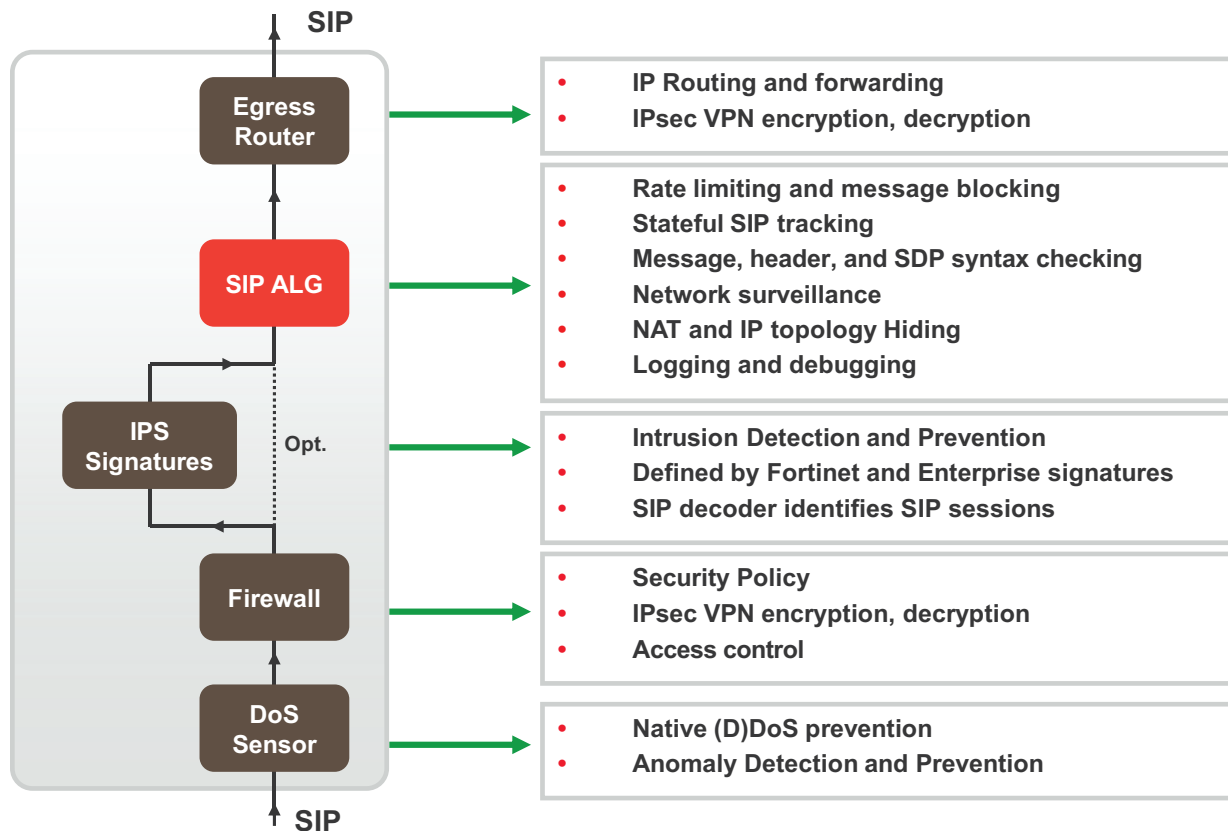
From the GUI you can only configure VoIP security profiles and add them to security policies if VoIP is turned on under **System > Feature Select** and if the **Inspection Mode** is set to **Proxy**. However, you can always configure VoIP profiles and add them to security profiles from the CLI. And if the `default-voip-alg` mode is set to `proxy-based` the default SIP profile will still be used even if VoIP security profiles are not visible from the GUI.

As shown in the figure below, the FortiGate SIP ALG intercepts SIP packets after they have been routed by the routing module, accepted by a security policy and passed through DoS and IPS Sensors (if DoS and IPS are enabled). The ALG raises SIP packets to the application layer, analyzes the SIP and SDP addressing information in the SIP messages, makes adjustments (for example, NAT) to this addressing if required, and then sends the packets out the egress interface to their destination.

The SIP ALG provides:

- All the same features as the SIP session helper including NAT and SIP and RTP Pinholes.
- In addition for the ALG you can enable or disable RTP pinholing, SIP register pinholing and SIP contact pinholing. In a signaling only environment where the RTP stream bypasses the FortiGate, you can disable RTP pinholing to improve performance.
- SIP TCP and UDP support
- SIP Message order checking
- Configurable Header line length maximums

The SIP ALG works at the application level after ingress packets are accepted by a security policy



- Message fragment assembly (TCP)
- If SIP messages are fragmented across multiple packets, the FortiGate assembles the fragments, does inspection and pass the message in its entirety to the SIP server as one packet. This offloads the server from doing all the TCP processing of fragments.
- L4 Protocol Translation
- Message Flood Protection
- Protects a SIP server from intentional or unintentional DoS of flooding INVITE, REGISTER, and other SIP methods by allowing control of the rate that these messages pass through the FortiGate.
- SIP message type filtering
- The FortiGate can prevent specified SIP message types from passing through the FortiGate to a SIP server. For example In a voice only SIP implementation, there may be no need to permit a SUBSCRIBE message to ever make it's way to the SIP call processor. Also, if a SIP server cannot process some SIP message types you can use SIP message type filtering to block them. For example, a SIP server could have a bug that prevents it from processing certain SIP messages. In this case you can temporarily block these message types until problem with the SIP server has been fixed.
- SIP statistics and logging
- SIP over IPv6
- SIP over SSL/TLS
- Deep SIP message syntax checking (also called deep SIP header inspection or SIP fuzzing protection). Prevents attacks that use malformed SIP messages. Can check many SIP headers and SDP statements. Configurable bypass and modification options.

- Hosted NAT traversal, Resolves IP address issue in SIP and SDP lines due to NAT-PT in far end firewall. Important feature for VoIP access networks.
- SIP High Availability (HA), including active-passive clustering and session pickup (session failover) for SIP sessions.
- Geographical Redundancy. In an HA configuration, if the active SIP server fails (missing SIP heartbeat messages or SIP traffic) SIP sessions can be redirected to a secondary SIP server in another location.
- SIP per request method message rate limitation with configurable threshold for SIP message rates per request method. Protects SIP servers from SIP overload and DoS attacks.
- RTP Bypass, Supports configurations with and without RTP pinholing. May inspect and protect SIP signaling only.
- SIP NAT with IP address conservation. Performs SIP and RTP aware IP Network Address translation. Preserves the lost IP address information in the SDP profile i= line for later processing/debugging in the SIP server. See [NAT with IP address conservation on page 82](#).
- IP topology hiding
- The IP topology of a network can be hidden through NAT and NAPT manipulation of IP and SIP level addressing. For example, see [SIP NAT scenario: source address translation \(source NAT\) on page 67](#).
- SIP inspection without address translation
- The SIP ALG inspects SIP messages but addresses in the messages are not translated. This feature can be applied to a FortiGate operating in Transparent mode or in NAT/Route mode. In Transparent mode you add normal Transparent mode security policies that enable the SIP ALG and include a VoIP profile that causes the SIP ALG to inspect SIP traffic as required. For an example configuration, see [Configuration example: SIP in Transparent Mode on page 56](#).
- For a FortiGate operating in NAT/Route mode, if SIP traffic can pass between different networks without requiring NAT because is supported by the routing configuration, you can add security policies that accept SIP traffic without enabling NAT. In the VoIP profile you can configure the SIP ALG to inspect SIP traffic as required.

## Enabling VoIP support from the GUI

Before you begin to configure VoIP security profiles, including SIP, from the GUI you should go to **System > Feature Select** and turn on **VoIP** (under **Additional Features**).

Also, VoIP settings are only available if the FortiGate or current VDOM **Inspection Mode** is set to **Proxy**. To view the inspection mode go to **System > Settings** to confirm that **Inspection Mode** is set to **Proxy**. You can also use the following CLI command to change the inspection mode to proxy:

```
config system settings
    set inspection-mode proxy
end
```

The **System Information** dashboard widget also shows the current **Mode**.

## SIP ALG configuration overview

To apply the SIP ALG, you add a SIP VoIP profile to a security policy that accepts SIP sessions. All SIP sessions accepted by the security policy will be processed by the SIP ALG using the settings in the VoIP profile. The VoIP profile contains settings that are applied to SIP, Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) and Skinny Call Control Protocol (SCCP) sessions. All SCCP sessions accepted by the security policy are also processed by the ALG. You configure SIP and SCCP settings separately. SIP settings also apply to SIMPLE sessions.

## VoIP profiles

You can customize the default VoIP profile or add new VoIP profiles.

To add a new VoIP profile from the GUI go to **Security Profiles > VoIP** and select **Create New** (the + button).

For SIP, from the GUI you can configure the VoIP profile to limit the number of SIP REGISTER and INVITE requests. Many additional options for configuring how the ALG processes SIP sessions are available from the CLI.

For SCCP you can limit the call setup time. Additional SCCP options are available from the CLI.

Use the following command to add a VoIP profile named VoIP\_Pro\_1 from the CLI:

```
config voip profile
  edit VoIP_Pro_1
end
```

FortiGates include two pre-defined VoIP profiles. On the GUI these profiles look identical. However, the CLI-only settings result in the following functionality.

|                |   |
|----------------|---|
| <b>default</b> | <p>The most commonly used VoIP profile. This profile enables both SIP and SCCP and places the minimum restrictions on what calls will be allowed to negotiate. This profile allows normal SCCP, SIP and RTP sessions and enables the following security settings:</p> <ul style="list-style-type: none"> <li>• <code>strict-register</code> to open smaller more secure pinholes (see <a href="#">Enhancing SIP pinhole security on page 88</a>).</li> <li>• <code>block-long-lines</code> to block SIP messages with lines that exceed maximum line lengths.</li> <li>• <code>block-unknown</code> to block unrecognized SIP request messages.</li> <li>• <code>open-record-route-pinhole</code> to open pinholes for Record-Route messages.</li> <li>• <code>nat-trace</code> (see <a href="#">NAT with IP address conservation on page 82</a>).</li> <li>• <code>contact-fixup</code> perform NAT on the IP addresses and port numbers in SIP headers in SIP CONTACT messages even if they don't match the session's IP address and port numbers.</li> <li>• <code>ips-rtp</code> to enable IPS in security policies that also accept SIP sessions to protect the SIP traffic from SIP-based attacks.</li> </ul> |
| <b>strict</b>  | <p>This profile is available for users who want to validate SIP messages and to only allow SIP sessions that are compliant with RFC 3261. In addition to the settings in the default VoIP profile, the strict profile sets all SIP deep message inspection header checking options (for example, <code>malformed-request-line</code> and many others) to <code>discard</code>. So the strict profile blocks and drops SIP messages that contain malformed SIP or SDP lines that can be detected by the ALG. For more information about SIP deep header inspection, see <a href="#">Deep SIP message inspection on page 98</a>.</p>  |

Neither of the default profiles applies SIP rate limiting. To apply more ALG features to SIP sessions you can clone (copy) the pre-defined VoIP profiles and make your own modifications to them. You can clone VoIP profiles from the GUI or the CLI. For example, from the CLI, to clone the default profile and configure the limit for SIP NOTIFY request messages to 1000 messages per second per security policy and block SIP INFO request messages.

```
config voip profile
  clone default to my_voip_pro
  edit my_voip_pro
    config sip
      set notify-rate 1000
      set block-info enable
    end
  end
```

## Changing the port numbers that the SIP ALG listens on

Most SIP configurations use TCP or UDP port 5060 for SIP sessions and port 5061 for SIP SSL sessions. If your SIP network uses different ports for SIP sessions you can use the following command to configure the SIP ALG to listen on a different TCP, UDP, or SSL ports. For example, to change the TCP port to 5064, the UDP port to 5065, and the SSL port to 5066.

```
config system settings
  set sip-tcp-port 5064
  set sip-udp-port 5065
  set sip-ssl-port 5066
end
```

You also configure the SIP ALG to listen in two different TCP ports and two different UDP ports for SIP sessions. For example, if you receive SIP TCP traffic on port 5060 and 5064 and UDP traffic on ports 5061 and 5065 you can enter the following command to receive the SIP traffic on all of these ports:

```
config system settings
  set sip-tcp-port 5060 5064
  set sip-udp-port 5061 5065
end
```

## Disabling the SIP ALG in a VoIP profile

SIP is enabled by default in a VoIP profile. If you are just using the VoIP profile for SCCP you can use the following command to disable SIP in the VoIP profile.

```
config voip profile
  edit VoIP_Pro_2
    config sip
      set status disable
    end
```

## SIP ALG diagnose commands

You can use the `diagnose sys sip-proxy` command to display diagnostic information for the SIP ALG. A number of options are available including:

Use the following command to list all active SIP calls being processed by the SIP ALG. You can also use the `clear` option to delete all active SIP calls being processed by the SIP ALG, the `idle` option to list idle SIP calls, and the `invite` option to list SIP invite transactions.

```
diagnose sys sip-proxy calls {clear | list | idle | invite}
```

Use the following commands to employ filters to display specific information about the SIP ALG and the session that it is processing. You can build up a filter by including a number of options such as source address, VoIP profile, policy, and so on.

```
diagnose sys sip-proxy filter <filter_options>
```

```
diagnose sys sip-proxy log-filter <filter_options>
```

Use the following command to display the active SIP rate limiting meters and their current settings.

```
diagnose sys sip-proxy meters list
```

Use the following command to display status information about the SIP sessions being processed by the SIP ALG. You can also clear all SIP ALG statistics.

```
diagnose sys sip-proxy stats {clear | list}
```



## Conflicts between the SIP ALG and the session helper

If you suspect that the SIP session helper is being used instead of the ALG, you can use the `diagnose sys sip` command to determine if the SIP session helper is processing SIP sessions. For example, the following command displays the overall status of the SIP sessions being processed by the SIP session helper:



The `diagnose sys sip` command only displays current status information. To see activity the SIP session helper has to actually be processing SIP sessions when you enter the command. For example, if the SIP session helper had been used for processing calls that ended 5 minutes ago, the command output would show no SIP session helper activity.

```
diagnose sys sip status
dialogs: max=32768, used=0
mappings: used=0
dialog hash by ID: size=2048, used=0, depth=0
dialog hash by RTP: size=2048, used=0, depth=0
mapping hash: size=2048, used=0, depth=0
count0: 0
count1: 0
count2: 0
count3: 0
count4: 0
```

This command output shows that the session helper is not processing SIP sessions because all of the used and count fields are 0. If any of these fields contains non-zero values then the SIP session helper may be processing SIP sessions.

Also, you can check to see if some ALG-only features are not being applied to all SIP sessions. For example, FortiView pages displays statistics for SIP and SCCP calls processed by the ALG but not for calls processed by the session helper. So if you see fewer calls than expected the session helper may be processing some of them.

Finally, you can check the policy usage and session information dashboard widgets to see if SIP sessions are being accepted by the wrong security policies.

## Stateful SIP tracking, call termination, and session inactivity timeout

The SIP ALG tracks SIP dialogs over their lifespan between the first INVITE message and the Final 200 OK and ACK messages. For every SIP dialog, stateful SIP tracking reviews every SIP message and makes adjustment to SIP tracking tables as required. These adjustments include source and destination IP addresses, address translation, dialog expiration information, and media stream port changes. Such changes can also result in dynamically opening and closing pinholes. You can use the `diagnose sys sip-proxy stats list` and the `diagnose sys sip-proxy filter` command to view the SIP call data being tracked by the SIP ALG.

The SIP ALG uses the SIP Expires header line to time out a SIP dialog if the dialog is idle and a Re-INVITE or UPDATE message is not received. The SIP ALG gets the Session-Expires value, if present, from the 200 OK response to the INVITE message. If the SIP ALG receives an INVITE before the session times out, all timeout values are reset to the settings in the new INVITE message or to default values. As a precautionary measure, the SIP ALG uses hard timeout values to set the maximum amount of time a call can exist. This ensures that the FortiGate is protected if a call ends prematurely.

When a SIP dialog ends normally, the SIP ALG deletes the SIP call information and closes open pinholes. A SIP call can also end abnormally due to an unexpected signaling or transport event that cuts off the call. When a call ends abnormally the SIP messages to end the call may not be sent or received. A call can end abnormally for the following reasons:

- Phones or servers crash during a call and a BYE message is not received.
- To attack a SIP system, a malicious user never send a BYE message.
- Poor implementations of SIP fail to process Record-Route messages and never send a BYE message.
- Network failures prevent a BYE message from being received.

Any phone or server in a SIP call can cancel the call by sending a CANCEL message. When a CANCEL message is received by the FortiGate, the SIP ALG closes open pinholes. Before terminating the call, the ALG waits for the final 200 OK message.

The SIP ALG can be configured to terminate SIP calls if the SIP dialog message flow or the call RTP (media) stream is interrupted and does not recover. You can use the following commands to configure terminating inactive SIP sessions and to set timers or counters to control when the call is terminated by the SIP ALG.

### Adding a media stream timeout for SIP calls

Use the following command in a VoIP profile to terminate SIP calls accepted by a security policy containing the VoIP profile when the RTP media stream is idle for 100 seconds.

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set call-keepalive 100
    end
  end
```

You can adjust this setting between 1 and 10,080 seconds. The default call keepalive setting of 0 disables terminating a call if the media stream is interrupted. Set call keepalive higher if your network has latency problems that could temporarily interrupt media streams. If you have configured call keepalive and the FortiGate terminates calls unexpectedly you can increase the call keepalive time to resolve the problem.



Call keep alive should be used with caution because enabling this feature results in extra FortiGate CPU overhead and can cause delay/jitter for the VoIP call. Also, the FortiGate terminates the call without sending SIP messages to end the call. And if the SIP endpoints send SIP messages to terminate the call they will be blocked by the FortiGate if they are sent after the FortiGate terminates the call.

## Adding an idle dialog setting for SIP calls

Use the following command in a VoIP profile to terminate SIP calls when for a single security policy, when the configured number of SIP calls (or dialogs) has stopped receiving SIP messages or has not received legitimate SIP messages. Using this command you can configure how many dialogs that have been accepted by a security policy that the VoIP profile is added to become idle before the SIP ALG deletes the oldest ones. The following command sets the maximum number of idle dialogs to 200:

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set max-idle-dialogs 200
    end
  end
```

Idle dialogs would usually be dialogs that have been interrupted because of errors or problems or as the result of a SIP attack that opens a large number of SIP dialogs without closing them. This command provides a way to remove these dialogs from the dialog table and recover memory and resources being used by these open and idle dialogs.

You can adjust this setting between 1 and a very high number. The default maximum idle dialogs setting of 0 disables this feature. Set maximum dialogs higher if your network has latency problems that could temporarily interrupt SIP messaging. If you have configured max idle dialogs and the FortiGate terminates calls unexpectedly you can increase the max idle dialogs number to resolve the problem.

## Changing how long to wait for call setup to complete

In some cases and some configurations your SIP system may experience delays during call setup. If this happens, some SIP ALG timers may expire before call setup is complete and drop the call. In some cases you may also want to reduce the amount of time the SIP ALG allows for call setup to complete.

You can use the `provisional-invite-expiry-time` SIP VoIP profile option to control how long the SIP ALG waits for provisional INVITE messages before assuming that the call setup has been interrupted and the SIP call should be dropped. The default value for this timer is 210 seconds. You can change it to between 10 and 3600 seconds.

Use the following command to change the expiry time to 100 seconds.

```
config voip profile
  edit Profile_name
```

```
    config sip
        set provisional-invite-expiry-time 100
    end
end
```

## SIP and RTP/RTCP

FortiGates support the Real Time Protocol (RTP) application layer protocol for the VoIP call audio stream. RTP uses dynamically assigned port numbers that can change during a call. SIP control messages that start a call and that are sent during the call inform callers of the port number to use and of port number changes during the call.

During a call, each RTP session will usually have a corresponding Real Time Control Protocol (RTCP) session. By default, the RTCP session port number is one higher than the RTP port number.

The RTP port number is included in the `m=` part of the SDP profile. In the example above, the SIP INVITE message includes RTP port number is 49170 so the RTCP port number would be 49171. In the SIP response message the RTP port number is 3456 so the RTCP port number would be 3457.

## How the SIP ALG creates RTP pinholes

The SIP ALG requires the following information to create a pinhole. The SIP ALG finds this information in SIP messages and some is provided by the SIP ALG:

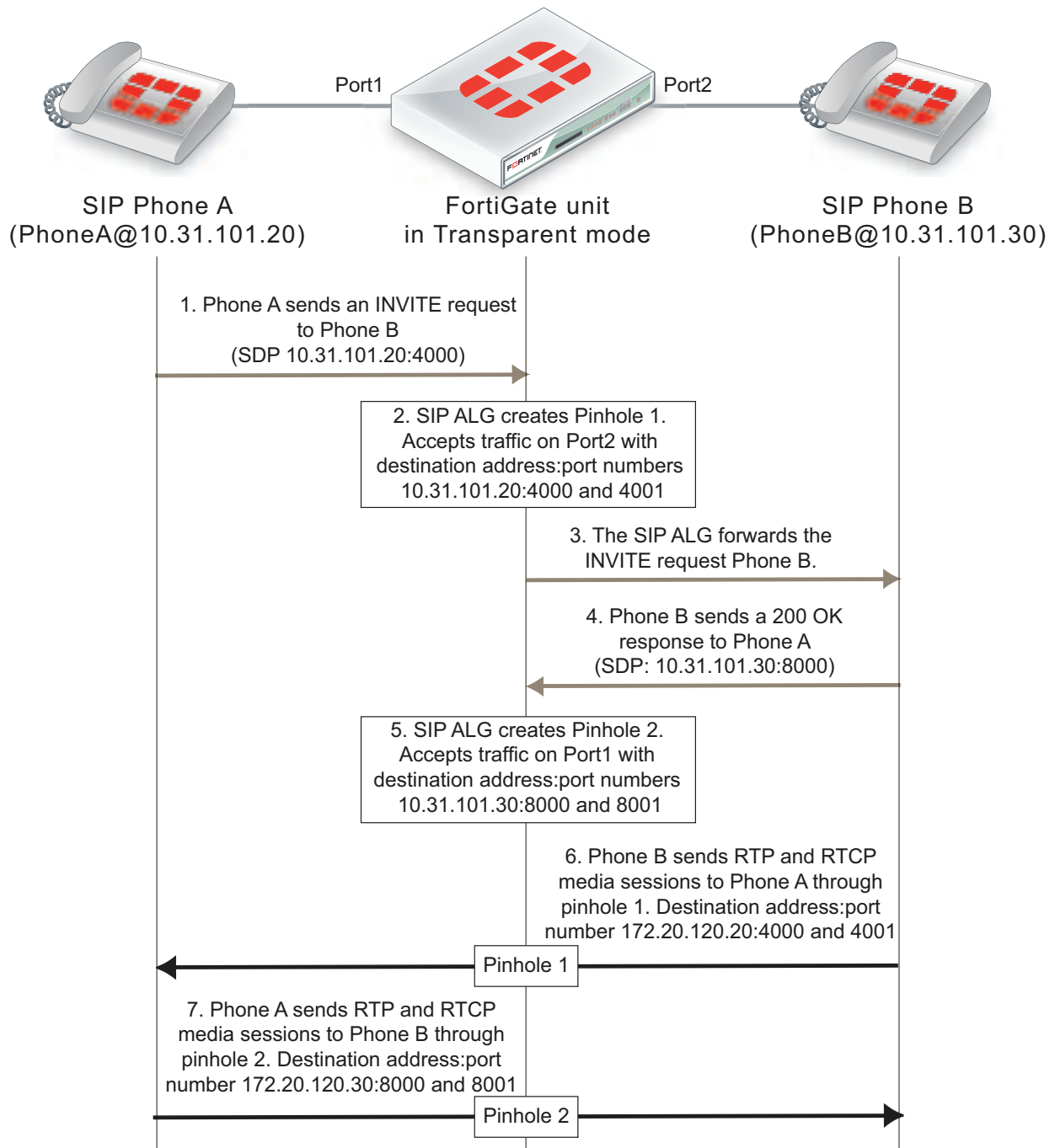
|                         |   |
|-------------------------|---|
| <b>Protocol</b>         | UDP (Extracted from SIP messages by the SIP ALG.)   |
| <b>Source IP</b>        | Any   |
| <b>Source port</b>      | Any   |
| <b>Destination IP</b>   | The SIP ALG extracts the destination IP address from the c= line in the SDP profile. The c= line can appear in either the session or media part of the SDP profile. The SIP ALG uses the IP address in the c= line of the media part of the SDP profile first. If the media part does not contain a c= line, the SIP ALG checks the c= line in the session part of the SDP profile. If the session part of the profile doesn't contain a c= line the packet is dropped. Pinholes for RTP and RTCP sessions share the same destination IP address. |
| <b>Destination port</b> | The SIP ALG extracts the destination port number for RTP from the m= field and adds 1 to this number to get the RTCP port number.   |
| <b>Lifetime</b>         | The length of time during which the pinhole will be open. When the lifetime ends, the SIP ALG removes the pinhole.  |

The SIP ALG keeps RTP pinholes open as long as the SIP session is alive. When the associated SIP session is terminated by the SIP ALG or the SIP phones or servers participating in the call, the RTP pinhole is closed.

The figure below shows a simplified call setup sequence that shows how the SIP ALG opens pinholes. Phone A and Phone B are installed on either side of a FortiGate operating in Transparent mode. Phone A and Phone B are on the same subnet. The FortiGate includes a security policy that accepts SIP sessions from port1 to port2 and from port2 to port1. The FortiGate does not require an RTP security policy, just the SIP policy.

You can see from this diagram that the SDP profile in the INVITE request from Phone A indicates that Phone A is expecting to receive a media stream sent to its IP address using port 4000 for RTP and port 4001 for RTCP. The SIP ALG creates pinhole 1 to allow this media traffic to pass through the FortiGate. Pinhole 1 is opened on the Port2 interface and will accept media traffic sent from Phone B to Phone A.

When Phone B receives the INVITE request from Phone A, Phone B will know to send media streams to Phone A using destination IP address 10.31.101.20 and ports 4000 and 4001. The 200 OK response sent from Phone B indicates that Phone B is expecting to receive a media stream sent to its IP address using ports 8000 and 8001. The SIP ALG creates pinhole 2 to allow this media traffic to pass through the FortiGate. Pinhole 2 is opened on the Port1 interface and will accept media traffic sent from Phone A to Phone B.

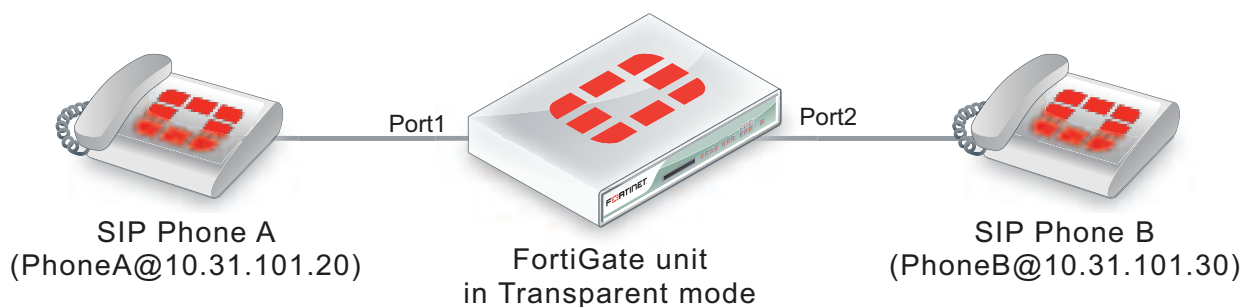
**SIP call setup with a FortiGate in Transparent mode**

## Configuration example: SIP in Transparent Mode

The figure below shows an example SIP network consisting of a FortiGate operating in Transparent mode between two SIP phones. Since the FortiGate is operating in Transparent mode both phones are on the same network and the FortiGate and the SIP ALG does not perform NAT. Even though the SIP ALG is not performing NAT you can use this configuration to apply SIP security features to the SIP traffic.

The FortiGate requires two security policies that accept SIP packets. One to allow SIP Phone A to start a session with SIP Phone B and one to allow SIP Phone B to start a session with SIP Phone A.

### SIP network with FortiGate in Transparent mode



## General configuration steps

The following general configuration steps are required for this SIP configuration. This example uses the default VoIP profile. The example also includes security policies that specifically allow SIP sessions using UDP port 5060 from Phone A to Phone B and from Phone B to Phone A. In most cases you would have more than two phones so would use more general security policies. Also, you can set the security service to ANY to allow traffic other than SIP on UDP port 5060.

1. Add firewall addresses for Phone A and Phone B.
2. Add a security policy that accepts SIP sessions initiated by Phone A and includes the default VoIP profile.
3. Add a security policy that accepts SIP sessions initiated by Phone B and includes the default VoIP profile.

## Configuration steps - GUI



Before you begin this procedure you may have to go to **System > Feature Select** and turn on VoIP.

### To add firewall addresses for the SIP phones

1. Go to **Policy & Objects > Addresses**.
2. Add the following addresses for Phone A and Phone B:



|                          |                              |
|--------------------------|------------------------------|
| <b>Category</b>          | Address                      |
| <b>Name</b>              | Phone_A                      |
| <b>Type</b>              | IP/Netmask                   |
| <b>Subnet / IP Range</b> | 10.31.101.20/255.255.255.255 |
| <b>Interface</b>         | port1                        |

|                          |                              |
|--------------------------|------------------------------|
| <b>Category</b>          | Address                      |
| <b>Name</b>              | Phone_B                      |
| <b>Type</b>              | IP/Netmask                   |
| <b>Subnet / IP Range</b> | 10.31.101.30/255.255.255.255 |
| <b>Interface</b>         | port2                        |

#### To add security policies to apply the SIP ALG to SIP sessions

1. Go to **Policy & Objects > IPv4 Policy**.
2. Add a security policy to allow Phone A to send SIP request messages to Phone B:

|                            |         |
|----------------------------|---------|
| <b>Incoming Interface</b>  | port1   |
| <b>Outgoing Interface</b>  | port2   |
| <b>Source</b>              | Phone_A |
| <b>Destination Address</b> | Phone_B |
| <b>Schedule</b>            | always  |
| <b>Service</b>             | SIP     |
| <b>Action</b>              | ACCEPT  |

3. Turn on **VoIP** and select the **default** VoIP profile.
4. Select **OK**.
5. Add a security policy to allow Phone B to send SIP request messages to Phone A:

|                            |         |
|----------------------------|---------|
| <b>Incoming Interface</b>  | port2   |
| <b>Outgoing Interface</b>  | port1   |
| <b>Source</b>              | Phone_B |
| <b>Destination Address</b> | Phone_A |

|                 |        |
|-----------------|--------|
| <b>Schedule</b> | always |
| <b>Service</b>  | SIP    |
| <b>Action</b>   | ACCEPT |

6. Turn on **VoIP** and select the **default** VoIP profile.
7. Select **OK**.

## Configuration steps - CLI

### To add firewall addresses for Phone A and Phone B and security policies to apply the SIP ALG to SIP sessions

1. Enter the following command to add firewall addresses for Phone A and Phone B.

```
config firewall address
edit Phone_A
set associated-interface port1
set type ipmask
set subnet 10.31.101.20 255.255.255.255
next
edit Phone_B
set associated-interface port2
set type ipmask
set subnet 10.31.101.30 255.255.255.255
end
```

2. Enter the following command to add security policies to allow Phone A to send SIP request messages to Phone B and Phone B to send SIP request messages to Phone A.

```
config firewall policy
edit 0
set srcintf port1
set dstintf port2
set srcaddr Phone_A
set dstaddr Phone_B
set action accept
set schedule always
set service SIP
set utm-status enable
set voip-profile default
next
edit 0
set srcintf port2
set dstintf port1
set srcaddr Phone_B
set dstaddr Phone_A
set action accept
set schedule always
set service SIP
set utm-status enable
set voip-profile default
end
```

## RTP enable/disable (RTP bypass)

You can configure the SIP ALG to stop from opening RTP pinholes. Called RTP bypass, this configuration can be used when you want to apply SIP ALG features to SIP signalling messages but do not want the RTP media streams to pass through the FortiGate. The FortiGate only acts as a signalling firewall and RTP media session bypass the FortiGate and no pinholes need to be created.

Enter the following command to enable RTP bypass in a VoIP profile by disabling opening RTP pinholes:

```
config voip profile
  edit VoIP_Pro_1
    config sip
      set rtp disable
    end
  end
end
```

## Opening and closing SIP register, contact, via and record-route pinholes

You can use the `open-register-pinhole`, `open-contact-pinhole`, `open-via-port`, and `open-record-route-pinhole` VoIP profile CLI options to control whether the FortiGate opens various pinholes.

If `open-register-pinhole` is enabled (the default setting) the FortiGate opens pinholes for SIP Register request messages. You can disable `open-register-pinhole` so that the FortiGate does not open pinholes for SIP Register request messages.

If `open-contact-pinhole` is enabled (the default setting) the FortiGate opens pinholes for non-Register SIP request messages. You can disable `open-contact-pinhole` so that the FortiGate does not open pinholes for non-register requests. Non-register pinholes are usually opened for SIP INVITE requests.

If `open-via-pinhole` is disabled (the default setting) the FortiGate does not open pinholes for Via messages. You can enable `open-via-pinhole` so that the FortiGate opens pinholes for Via messages.

If `open-record-route-pinhole` is enabled (the default setting) the FortiGate opens pinholes for Record-Route messages. You can disable `open-record-route-pinhole` so that the FortiGate does not open pinholes for Record-Route messages.

Usually you would want to open these pinholes. Keeping them closed may prevent SIP from functioning properly through the FortiGate. They can be disabled, however, for interconnect scenarios (where all SIP traffic is between proxies and traveling over a single session). In some cases these settings can also be disabled in access scenarios if it is known that all users will be registering regularly so that their contact information can be learned from the register request.

You might want to prevent pinholes from being opened to avoid creating a pinhole for every register or non-register request. Each pinhole uses additional system memory, which can affect system performance if there are hundreds or thousands of users, and requires refreshing which can take a relatively long amount of time if there are thousands of active calls.

To configure a VoIP profile to prevent opening register and non-register pinholes:

```
config voip profile
  edit VoIP_Pro_1
    config sip
      set open-register-pinhole disable
      set open-contact-pinhole disable
    end
  end
```

In some cases you may not want to open pinholes for the port numbers specified in SIP Contact headers. For example, in an interconnect scenario when a FortiGate is installed between two SIP servers and the only SIP traffic through the FortiGate is between these SIP servers pinholes may not need to be opened for the port numbers specified in the Contact header lines.

If you disable `open-register-pinhole` then pinholes are not opened for ports in Contact header lines in SIP Register messages. If you disable `open-contact-pinhole` then pinholes are not opened for ports in Contact header lines in all SIP messages except SIP Register messages.

## Accepting SIP register responses

You can enable the VoIP profile `open-via-pinhole` options to accept a SIP Register response message from a SIP server even if the source port of the Register response message is different from the destination port.

Most SIP servers use 5060 as the source port in the SIP register response. Some SIP servers, however, may use a different source port. If your SIP server uses a different source port, you can enable `open-via-pinhole` and the SIP ALG will create a temporary pinhole when the Register request from a SIP client includes a different source port. The FortiGate will accept a SIP Register response with any source port number from the SIP server.

Enter the following command to enable accepting any source port from a SIP server:

```
config voip profile
  edit VoIP_Pro_1
    config sip
      set open-via-pinhole enable
    end
  end
end
```

## How the SIP ALG performs NAT

In most Network Address Translation (NAT) configurations, multiple hosts in a private network share a single public IP address to access the Internet. For sessions originating on the private network for the Internet, NAT replaces the private IP address of the PC in the private subnet with the public IP address of the NAT device. The NAT device converts the public IP address for responses from the Internet back into the private address before sending the response over the private network to the originator of the session.

Using NAT with SIP is more complex because of the IP addresses and media stream port numbers used in SIP message headers and bodies. When a caller on the private network sends a SIP message to a phone or SIP server on the Internet, the SIP ALG must translate the private network addresses in the SIP message to IP addresses and port numbers that are valid on the Internet. When the response message is sent back to the caller, the SIP ALG must translate these addresses back to valid private network addresses.

In addition, the media streams generated by the SIP session are independent of the SIP message sessions and use varying port numbers that can also change during the media session. The SIP ALG opens pinholes to accept these media sessions, using the information in the SIP messages to determine the pinholes to open. The ALG may also perform port translation on the media sessions.

When an INVITE message is received by the SIP ALG, the FortiGate extracts addressing and port number information from the message header and stores it in a SIP dialog table. Similar to an IP session table the data in the dialog table is used to translate addresses in subsequent SIP messages that are part of the same SIP call.

When the SIP ALG receives a response to the INVITE message arrives, (for example, an ACK or 200 OK), the SIP ALG compares the addresses in the message fields against the entries in the SIP dialog table to identify the call context of the message. The SIP ALG then translates addresses in the SIP message before forwarding them to their destination.

The addressing and port number information in SDP fields is used by the ALG to reserve ports for the media session and create a NAT mapping between them and the ports in the SDP fields. Because SDP uses sequential ports for the RTP and RTCP channels, the ALG provides consecutive even-odd ports.

## SIP ALG source address translation

When a SIP call is started by a phone on a private network destined for a phone on the Internet, only source address translation is required. The phone on the private network attempts to contact the actual IP address of the phone on the Internet. However, the source address of the phone on the private network is not routable on the Internet so the SIP ALG must translate all private IP addresses in the SIP message into public IP addresses.

To configure the FortiGate for source address translation you add security policy that accepts sessions from the internal network destined for the Internet. You must enable NAT for the security policy and add a VoIP profile.

When a SIP request is received from the internal to the external network, the SIP ALG replaces the private network IP addresses and port numbers in the SIP message with the IP address of the FortiGate interface connected to the Internet. Depending on the content of the message, the ALG translates addresses in the Via:, Contact:, Route:, and Record-Route: SIP header fields. The message is then forwarded to the destination (either a VoIP phone or a SIP server on the Internet).

The VoIP phone or server in the Internet sends responses to these SIP messages to the external interface of the FortiGate. The addresses in the response messages are translated back into private network addresses and the response is forwarded to the originator of the request.

For the RTP communication between the SIP phones, the SIP ALG opens pinholes to allow media through the FortiGate on the dynamically assigned ports negotiated based on information in the SDP and the Via:, Contact:, and Record-Route: header fields. The pinholes also allow incoming packets to reach the Contact:, Via:, and Record-Route: IP addresses and ports. When processing return traffic, the SIP ALG inserts the original Contact:, Via:, Route:, and Record-Route: SIP fields back into the packets.

## SIP ALG destination address translation

Incoming calls are directed from a SIP phone on the Internet to the interface of the FortiGate connected to the Internet. To receive these calls you must add a security policy to accept SIP sessions from the Internet. The security policy requires a firewall virtual IP. SIP INVITE messages from the Internet connect to the external IP address of the virtual IP. The SIP ALG uses the destination address translation defined in the virtual IP to translated the addresses in the SIP message to addresses on the private network.

When a 200 OK response message arrives from the private network, the SIP ALG translates the addresses in the message to Internet addresses and opens pinholes for media sessions from the private network to the Internet.

When the ACK message is received for the 200 OK, it is also intercepted by the SIP ALG. If the ACK message contains SDP information, the SIP ALG checks to determine if the IP addresses and port numbers are not changed from the previous INVITE. If they are, the SIP ALG deletes pinholes and creates new ones as required. The ALG also monitors the Via:, Contact:, and Record-Route: SIP fields and opens new pinholes as required.

## SIP Call Re-invite messages

SIP Re-INVITE messages can dynamically add and remove media sessions during a call. When new media sessions are added to a call the SIP ALG opens new pinholes and update SIP dialog data. When media sessions are ended, the SIP ALG closes pinholes that are no longer needed and removes SIP dialog data.

## How the SIP ALG translates IP addresses in SIP headers

The SIP ALG applies NAT to SIP sessions by translating the IP addresses contained in SIP headers. For example, the following SIP message contains most of the SIP fields that contain addresses that need to be translated:

```
INVITE PhoneB@172.20.120.30 SIP/2.0
Via: SIP/2.0/UDP 172.20.120.50:5434
From: PhoneA@10.31.101.20
To: PhoneB@172.20.120.30
Call-ID: a12abcde@172.20.120.50
Contact: PhoneA@10.31.101.20:5434
Route: <sip:example@172.20.120.50:5060>
Record-Route: <sip:example@172.20.120.50:5060>
```

How IP address translation is performed depends on whether source NAT or destination NAT is applied to the session containing the message:

## Source NAT translation of IP addresses in SIP messages

Source NAT translation occurs for SIP messages sent from a phone or server on a private network to a phone or server on the Internet. The source addresses in the SIP header fields of the message are typically set to IP addresses on the private network. The SIP ALG translates these addresses to the address the FortiGate interface connected to the Internet.

### Source NAT translation of IP addresses in SIP request messages

| SIP header           | NAT action  |
|----------------------|---|
| <b>To:</b>           | None  |
| <b>From:</b>         | Replace private network address with IP address of FortiGate interface connected to the Internet. |
| <b>Call-ID:</b>      | Replace private network address with IP address of FortiGate interface connected to the Internet. |
| <b>Via:</b>          | Replace private network address with IP address of FortiGate interface connected to the Internet. |
| <b>Request-URI:</b>  | None  |
| <b>Contact:</b>      | Replace private network address with IP address of FortiGate interface connected to the Internet. |
| <b>Record-Route:</b> | Replace private network address with IP address of FortiGate interface connected to the Internet. |
| <b>Route:</b>        | Replace private network address with IP address of FortiGate interface connected to the Internet. |

Response messages from phones or servers on the Internet are sent to the FortiGate interface connected to the Internet where the destination addresses are translated back to addresses on the private network before forwarding the SIP response message to the private network.

### Source NAT translation of IP addresses in SIP response messages

| SIP header   | NAT action  |
|--------------|---|
| <b>To:</b>   | None  |
| <b>From:</b> | Replace IP address of FortiGate interface connected to the Internet with private network address. |



| SIP header           | NAT action  |
|----------------------|---|
| <b>Call-ID:</b>      | Replace IP address of FortiGate interface connected to the Internet with private network address. |
| <b>Via:</b>          | Replace IP address of FortiGate interface connected to the Internet with private network address. |
| <b>Request-URI:</b>  | N/A   |
| <b>Contact:</b>      | None  |
| <b>Record-Route:</b> | Replace IP address of FortiGate interface connected to the Internet with private network address. |
| <b>Route:</b>        | Replace IP address of FortiGate interface connected to the Internet with private network address. |

## Destination NAT translation of IP addresses in SIP messages

Destination NAT translation occurs for SIP messages sent from a phone or server on the Internet to a firewall virtual IP address. The destination addresses in the SIP header fields of the message are typically set to the virtual IP address. The SIP ALG translates these addresses to the address of a SIP server or phone on the private network on the other side of the FortiGate.

### Destination NAT translation of IP addresses in SIP request messages

| SIP header           | NAT action   |
|----------------------|--|
| <b>To:</b>           | Replace VIP address with address on the private network as defined in the firewall virtual IP. |
| <b>From:</b>         | None   |
| <b>Call-ID:</b>      | None   |
| <b>Via:</b>          | None   |
| <b>Request-URI:</b>  | Replace VIP address with address on the private network as defined in the firewall virtual IP. |
| <b>Contact:</b>      | None   |
| <b>Record-Route:</b> | None   |
| <b>Route:</b>        | None   |

SIP response messages sent in response to the destination NAT translated messages are sent from a server or a phone on the private network back to the originator of the request messages on the Internet. These reply

messages are accepted by the same security policy that accepted the initial request messages, The firewall VIP in the original security policy contains the information that the SIP ALG uses to translate the private network source addresses in the SIP headers into the firewall virtual IP address.

### Destination NAT translation of IP addresses in SIP response messages

| SIP header    | NAT action   |
|---------------|--|
| To:           | None   |
| From:         | Replace private network address with firewall VIP address. |
| Call-ID:      | None   |
| Via:          | None   |
| Request-URI:  | N/A  |
| Contact:      | Replace private network address with firewall VIP address. |
| Record-Route: | Replace private network address with firewall VIP address. |
| Route:        | None   |

## How the SIP ALG translates IP addresses in the SIP body

The SDP session profile attributes in the SIP body include IP addresses and port numbers that the SIP ALG uses to create pinholes for the media stream.

The SIP ALG translates IP addresses and port numbers in the o=, c=, and m= SDP lines. For example, in the following lines the ALG could translate the IP addresses in the o= and c= lines and the port number (49170) in the m= line.

```
o=PhoneA 5462346 332134 IN IP4 10.31.101.20
c=IN IP4 10.31.101.20
m=audio 49170 RTP 0 3
```

If the SDP session profile includes multiple RTP media streams, the SIP ALG opens pinholes and performs the required address translation for each one.

The two most important SDP attributes for the SIP ALG are c= and m=. The c= attribute is the connection information attribute. This field can appear at the session or media level. The syntax of the connection attribute is:

```
c=IN {IPv4 | IPv6} <destination_ip_address>
```

Where

- **IN** is the network type. FortiGates support the **IN** or Internet network type.
- **{IPv4 | IPv6}** is the address type. FortiGates support IPv4 or IPv6 addresses in SDP statements. However, FortiGates do not support all types of IPv6 address translation. See [SIP over IPv6 on page 97](#).
- **<destination\_IP\_address>** is the unicast numeric destination IP address or domain name of the connection in either IPv4 or IPv6 format.

The syntax of the media attribute is:

```
m=audio <port_number> RTP <format_list>
```

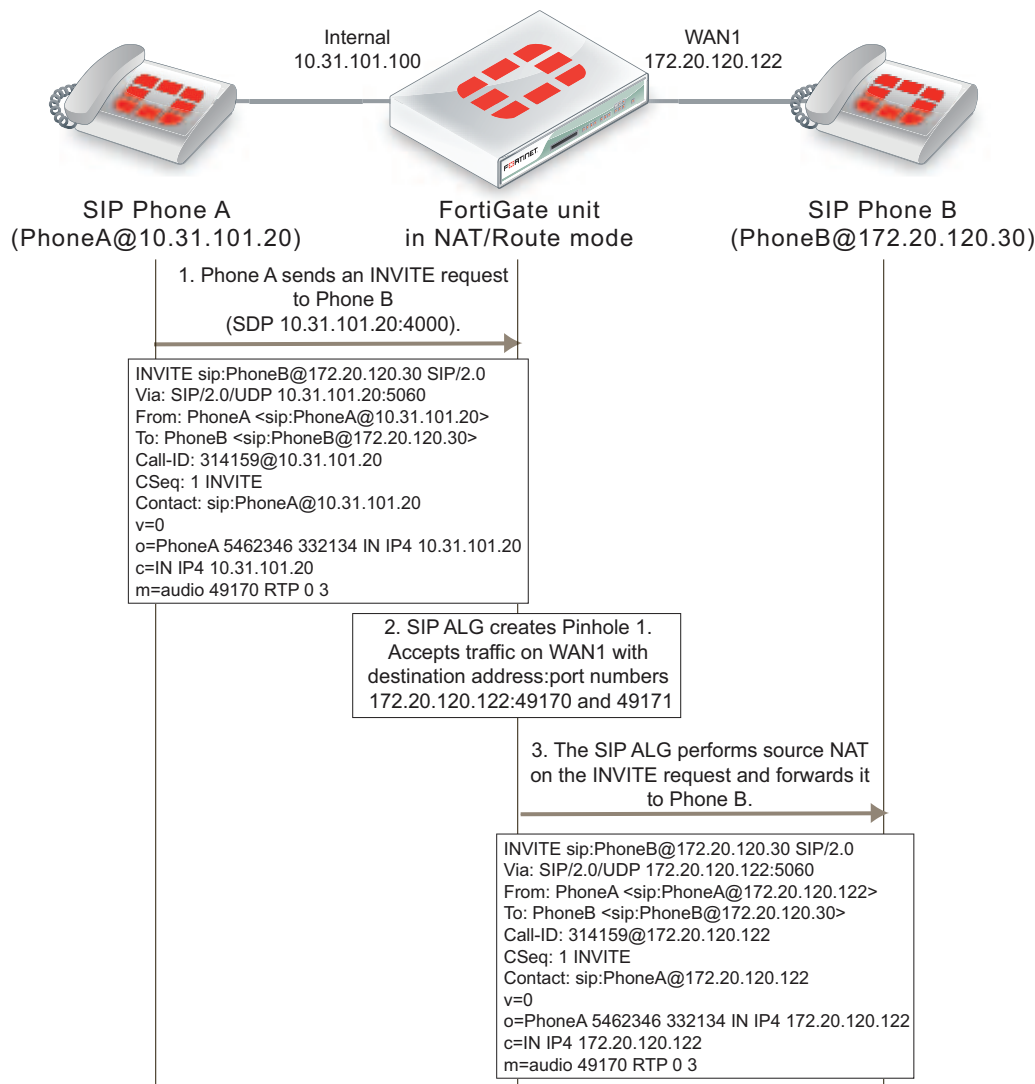
Where

- `audio` is the media type. FortiGates support the `audio` media type.
- `<port_number>` is the destination port number used by the media stream.
- `RTP` is the application layer transport protocol used for the media stream. FortiGates support the Real Time Protocol (RTP) transport protocol.
- `<format_list>` is the format list that provides information about the application layer protocol that the media uses.

## SIP NAT scenario: source address translation (source NAT)

The following figures show a source address translation scenario involving two SIP phones on different networks, separated by a FortiGate. In the scenario, SIP Phone A sends an INVITE request to SIP Phone B and SIP Phone B replies with a 200 OK response and then the two phones start media streams with each other.

To simplify the diagrams, some SIP messages are not included (for example, the Ringing and ACK response messages) and some SIP header lines and SDP profile lines have been removed from the SIP messages.

**SIP source NAT scenario part 1: INVITE request sent from Phone A to Phone B**

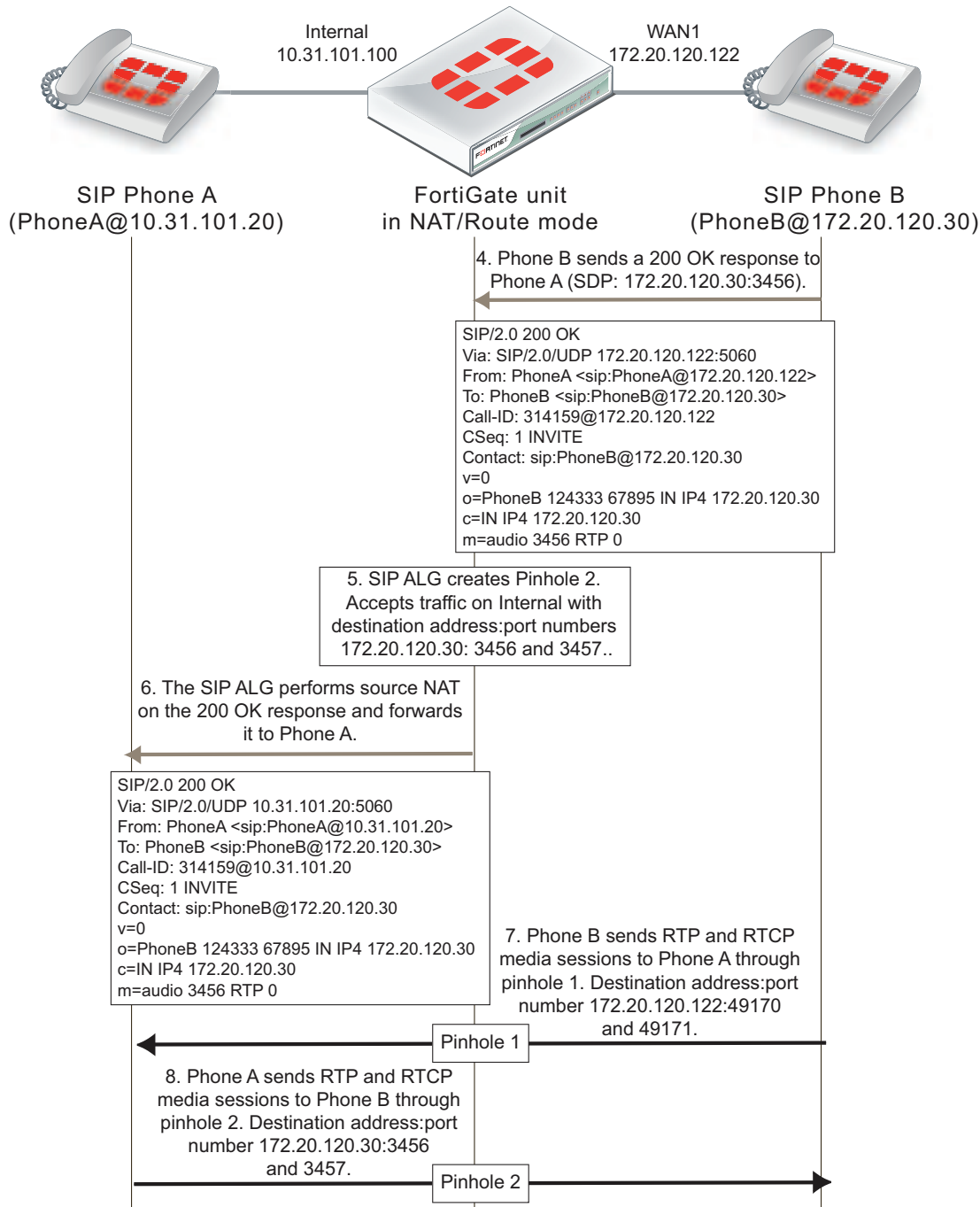
For the replies to SIP packets sent by Phone A to be routable on Phone B's network, the FortiGate uses source NAT to change their source address to the address of the WAN1 interface. The SIP ALG makes similar changes to the source addresses in the SIP headers and SDP profile. For example, the original INVITE request from Phone A includes the address of Phone A (10.31.101.20) in the from header line. After the INVITE request passes through the FortiGate, the address of Phone A in the From SIP header line is translated to 172.20.120.122, the address of the FortiGate WAN1 interface. As a result, Phone B will reply to SIP messages from Phone A using the WAN1 interface IP address.

The FortiGate also opens a pinhole so that it can accept media sessions sent to the WAN1 IP address using the port number in the m= line of the INVITE request and forward them to Phone A after translating the destination address to the IP address of Phone A.

Phone B sends the 200 OK response to the INVITE message to the WAN1 interface. The SDP profile includes the port number that Phone B wants to use for its media stream. The FortiGate forwards 200 OK response to Phone A after translating the addresses in the SIP and SDP lines back to the IP address of Phone A. The SIP ALG also

opens a pinhole on the Internal interface that accepts media stream sessions from Phone A with destination address set to the IP address of Phone B and using the port that Phone B added to the SDP m= line.

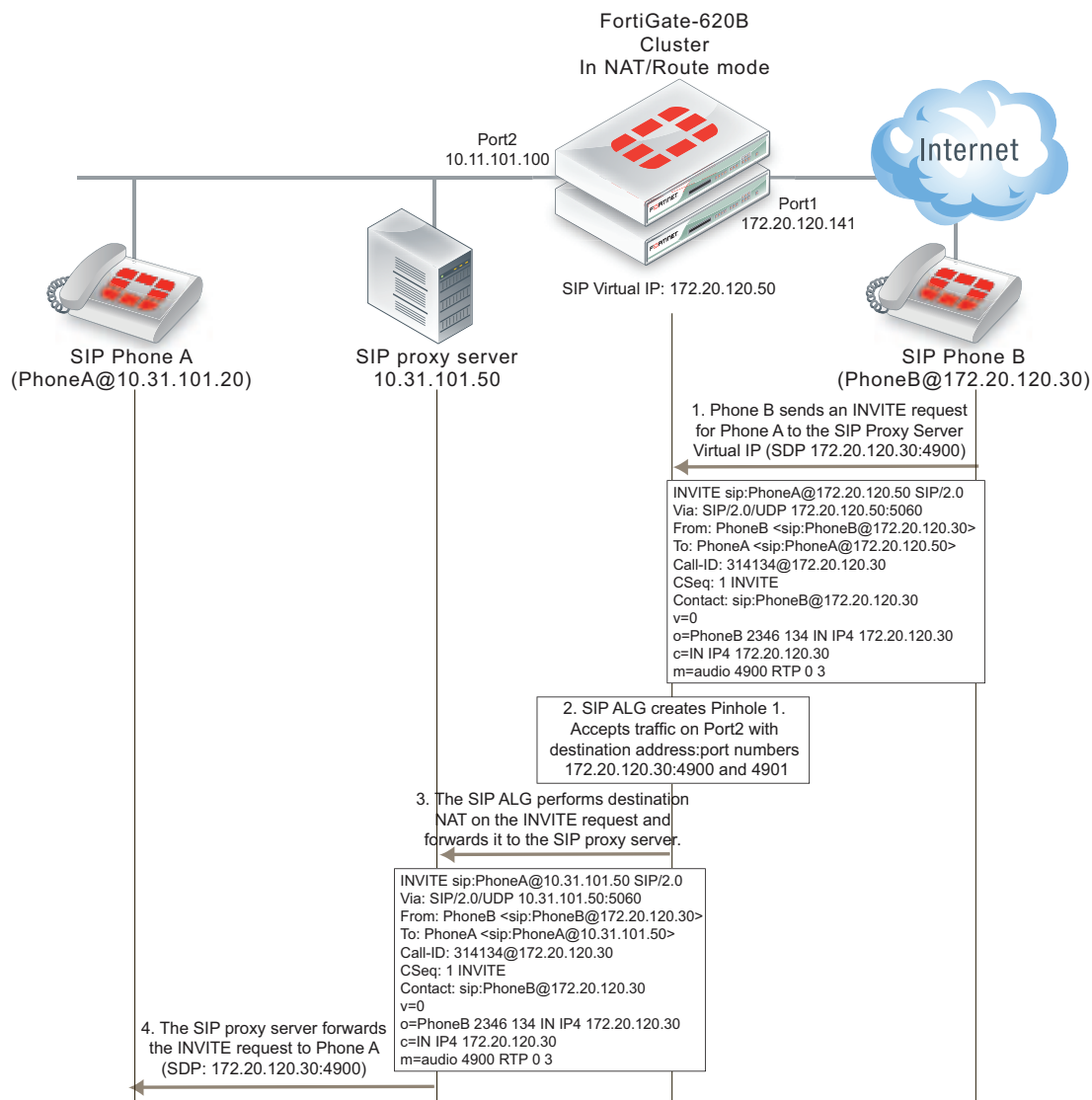
### SIP source NAT scenario part 2: 200 OK returned and media streams established



## SIP NAT scenario: destination address translation (destination NAT)

The following figures show how the SIP ALG translates addresses in a SIP INVITE message sent from SIP Phone B on the Internet to SIP Phone A on a private network using the SIP proxy server. Because the addresses on the private network are not visible from the Internet, the security policy on the FortiGate that accepts SIP sessions includes a virtual IP. Phone A sends SIP the INVITE message to the virtual IP address. The FortiGate accepts the INVITE message packets and using the virtual IP, translates the destination address of the packet to the IP address of the SIP proxy server and forwards the SIP message to it.

### SIP destination NAT scenario part 1: INVITE request sent from Phone B to Phone A

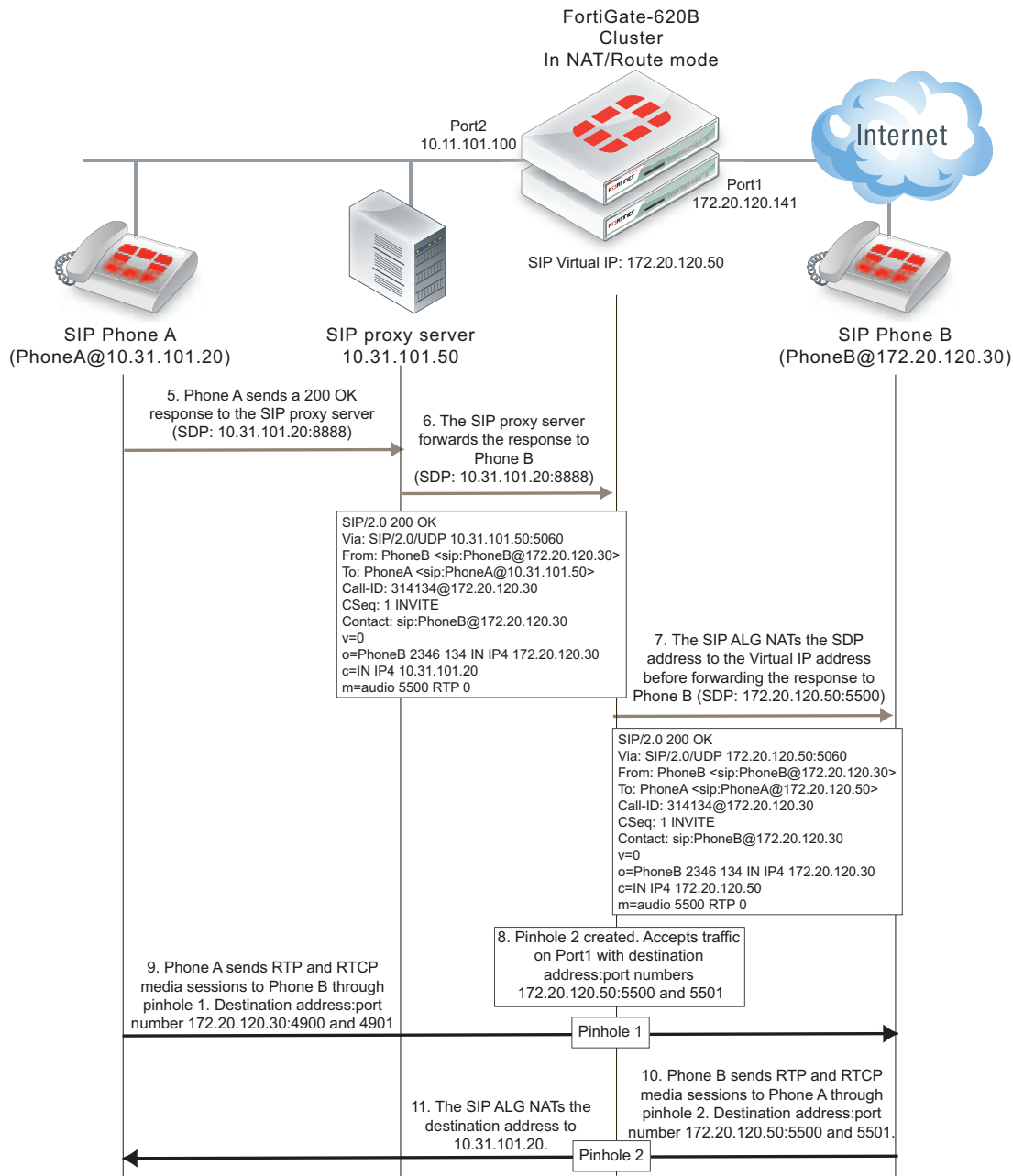


To simplify the diagrams, some SIP messages are not included (for example, the Ringing and ACK response messages) and some SIP header lines and SDP profile lines have been removed from the SIP messages.

The SIP ALG also translates the destination addresses in the SIP message from the virtual IP address (172.20.120.50) to the SIP proxy server address (10.31.101.50). For this configuration to work, the SIP proxy server must be able to change the destination addresses for Phone A in the SIP message from the address of the SIP proxy server to the actual address of Phone A.

The SIP ALG also opens a pinhole on the Port2 interface that accepts media sessions from the private network to SIP Phone B using ports 4900 and 4901.

Phone A sends a 200 OK response back to the SIP proxy server. The SIP proxy server forwards the response to Phone B. The FortiGate accepts the 100 OK response. The SIP ALG translates the Phone A addresses back to the SIP proxy server virtual IP address before forwarding the response back to Phone B. The SIP ALG also opens a pinhole using the SIP proxy server virtual IP which is the address in the o= line of the SDP profile and the port number in the m= line of the SDP code.

**SIP destination NAT scenario part 2: 200 OK returned to Phone B and media streams established**

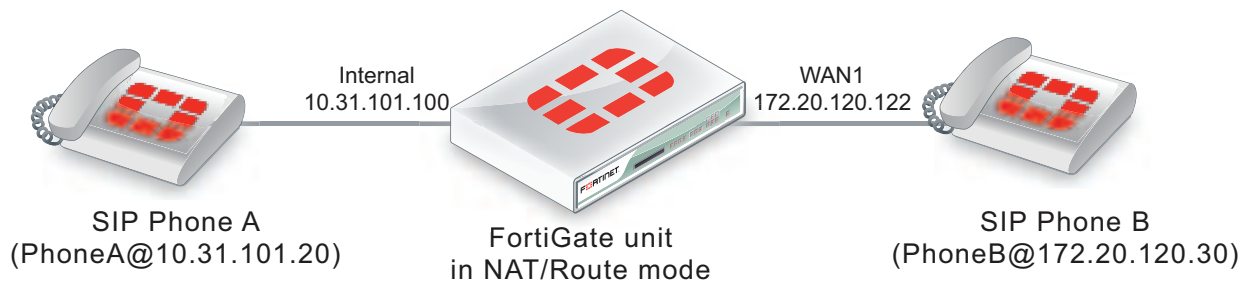
The media stream from Phone A is accepted by pinhole one and forwarded to Phone B. The source address of this media stream is changed to the SIP proxy server virtual IP address. The media stream from Phone B is accepted by pinhole 2 and forwarded to Phone B. The destination address of this media stream is changed to the IP address of Phone A.



## SIP NAT configuration example: source address translation (source NAT)

This configuration example shows how to configure the FortiGate to support the source address translation scenario shown below. The FortiGate requires two security policies that accept SIP packets. One to allow SIP Phone A to start a session with SIP Phone B and one to allow SIP Phone B to start a session with SIP Phone A. Both of these policies must include source NAT. In this example the networks are not hidden from each other so destination NAT is not required.

### SIP source NAT configuration



### General configuration steps

The following general configuration steps are required for this SIP configuration. This example uses the default VoIP profile. The example also includes security policies that specifically allow SIP sessions using UDP port 5060 from Phone A to Phone B and from Phone B to Phone A. In most cases you would have more than two phones so would use more general security policies. Also, you can set the firewall service to ANY to allow traffic other than SIP on UDP port 5060.

1. Add firewall addresses for Phone A and Phone B.
2. Add a security policy that accepts SIP sessions initiated by Phone A and includes the default VoIP profile.
3. Add a security policy that accepts SIP sessions initiated by Phone B and includes the default VoIP profile.

## Configuration steps - GUI

### To add firewall addresses for the SIP phones

1. Go to **Policy & Objects > Addresses**.
2. Add the following addresses for Phone A and Phone B:

| Category          | Address                      |
|-------------------|------------------------------|
| Name              | Phone_A                      |
| Type              | IP/Netmask                   |
| Subnet / IP Range | 10.31.101.20/255.255.255.255 |
| Interface         | Internal                     |

|                          |                               |
|--------------------------|-------------------------------|
| <b>Category</b>          | Address                       |
| <b>Name</b>              | Phone_B                       |
| <b>Type</b>              | IP/Netmask                    |
| <b>Subnet / IP Range</b> | 172.20.120.30/255.255.255.255 |
| <b>Interface</b>         | wan1                          |

**To add security policies to apply the SIP ALG to SIP sessions**

1. Go to **Policy & Objects > Policy > IPv4**.
2. Add a security policy to allow Phone A to send SIP request messages to Phone B:

|                            |          |
|----------------------------|----------|
| <b>Incoming Interface</b>  | internal |
| <b>Outgoing Interface</b>  | wan1     |
| <b>Source</b>              | Phone_A  |
| <b>Destination Address</b> | Phone_B  |
| <b>Schedule</b>            | always   |
| <b>Service</b>             | SIP      |
| <b>Action</b>              | ACCEPT   |

3. Turn on **NAT** and select **Use Outgoing Interface Address**.
4. Turn on **VoIP** and select the **default** VoIP profile.
5. Select **OK**.
6. Add a security policy to allow Phone B to send SIP request messages to Phone A:

|                            |          |
|----------------------------|----------|
| <b>Incoming Interface</b>  | wan1     |
| <b>Outgoing Interface</b>  | internal |
| <b>Source</b>              | Phone_B  |
| <b>Destination Address</b> | Phone_A  |
| <b>Schedule</b>            | always   |
| <b>Service</b>             | SIP      |
| <b>Action</b>              | ACCEPT   |

7. Turn on **NAT** and select **Use Outgoing Interface Address**.
8. Turn on **VoIP** and select the **default** VoIP profile.
9. Select **OK**.

## Configuration steps - CLI

### To add firewall addresses for Phone A and Phone B and security policies to apply the SIP ALG to SIP sessions

1. Enter the following command to add firewall addresses for Phone A and Phone B.

```
config firewall address
edit Phone_A
set associated interface internal
set type ipmask
set subnet 10.31.101.20 255.255.255.255
next
edit Phone_B
set associated interface wan1
set type ipmask
set subnet 172.20.120.30 255.255.255.255
end
```

2. Enter the following command to add security policies to allow Phone A to send SIP request messages to Phone B and Phone B to send SIP request messages to Phone A.

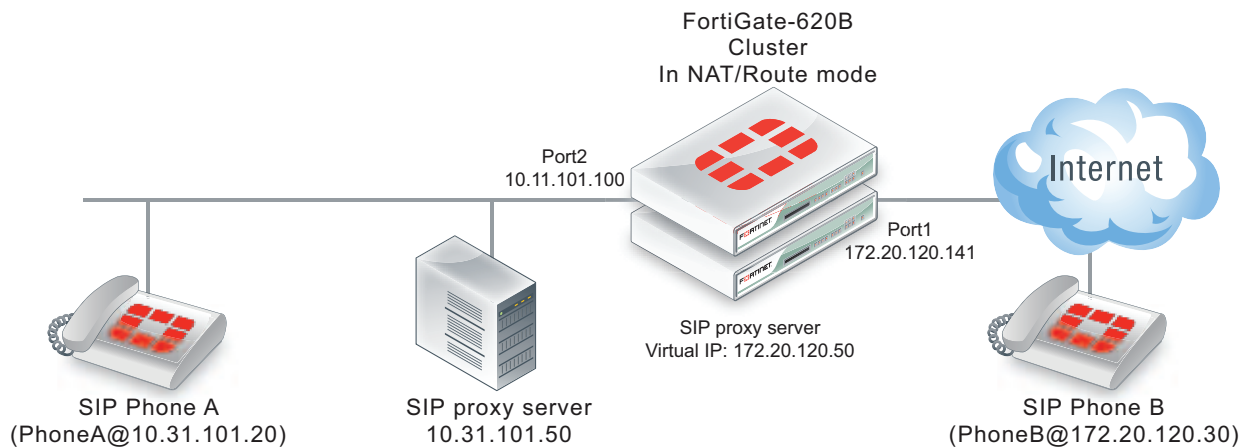
```
config firewall policy
edit 0
set srcintf internal
set dstintf wan1
set srcaddr Phone_A
set dstaddr Phone_B
set action accept
set schedule always
set service SIP
set nat enable
set utm-status enable
set voip-profile default
next
edit 0
set srcintf wan1
set dstintf internal
set srcaddr Phone_B
set dstaddr Phone_A
set action accept
set schedule always
set service SIP
set nat enable
set utm-status enable
set voip-profile default
end
```

## SIP NAT configuration example: destination address translation (destination NAT)

This configuration example shows how to configure the FortiGate to support the destination address translation scenario shown in the figure below. The FortiGate requires two SIP security policies:

- A destination NAT security policy that allows SIP messages to be sent from the Internet to the private network. This policy must include destination NAT because the addresses on the private network are not routable on the Internet.
- A source NAT security policy that allows SIP messages to be sent from the private network to the Internet.

### SIP destination NAT scenario part two: 200 OK returned to Phone B and media streams established



## General configuration steps

The following general configuration steps are required for this destination NAT SIP configuration. This example uses the default VoIP profile.

1. Add the SIP proxy server firewall virtual IP.
2. Add a firewall address for the SIP proxy server on the private network.
3. Add a destination NAT security policy that accepts SIP sessions from the Internet destined for the SIP proxy server virtual IP and translates the destination address to the IP address of the SIP proxy server on the private network.
4. Add a security policy that accepts SIP sessions initiated by the SIP proxy server and destined for the Internet.

## Configuration steps - GUI

### To add the SIP proxy server firewall virtual IP

1. Go to **Policy & Objects > Virtual IPs**.
2. Add the following SIP proxy server virtual IP.

|          |      |
|----------|------|
| VIP Type | IPv4 |
|----------|------|

|                                  |               |
|----------------------------------|---------------|
| <b>Name</b>                      | SIP_Proxy_VIP |
| <b>Interface</b>                 | port1         |
| <b>Type</b>                      | Static NAT    |
| <b>External IP Address/Range</b> | 172.20.120.50 |
| <b>Mapped IP Address/Range</b>   | 10.31.101.50  |

#### To add a firewall address for the SIP proxy server

1. Go to **Policy & Objects > Addresses**.
2. Add the following for the SIP proxy server:

|                        |                              |
|------------------------|------------------------------|
| <b>Address Name</b>    | SIP_Proxy_Server             |
| <b>Type</b>            | Subnet                       |
| <b>Subnet/IP Range</b> | 10.31.101.50/255.255.255.255 |
| <b>Interface</b>       | port2                        |

#### To add the security policies

1. Go to **Policy & Objects > IPv4 Policy**.
2. Add a destination NAT security policy that includes the SIP proxy server virtual IP that allows Phone B (and other SIP phones on the Internet) to send SIP request messages to the SIP proxy server.

|                            |               |
|----------------------------|---------------|
| <b>Incoming Interface</b>  | port1         |
| <b>Outgoing Interface</b>  | port2         |
| <b>Source</b>              | all           |
| <b>Destination Address</b> | SIP_Proxy_VIP |
| <b>Schedule</b>            | always        |
| <b>Service</b>             | SIP           |
| <b>Action</b>              | ACCEPT        |

3. Turn on **NAT** and select **Use Outgoing Interface Address**.
4. Turn on **VoIP** and select the **default** VoIP profile.
5. Select **OK**.
6. Add a source NAT security policy to allow the SIP proxy server to send SIP request messages to Phone B and the Internet:

|                           |       |
|---------------------------|-------|
| <b>Incoming Interface</b> | port2 |
|---------------------------|-------|

|                            |                  |
|----------------------------|------------------|
| <b>Destination Address</b> | all              |
| <b>Source</b>              | SIP_Proxy_Server |
| <b>Schedule</b>            | always           |
| <b>Service</b>             | SIP              |
| <b>Action</b>              | ACCEPT           |

7. Turn on **NAT** and select **Use OutgoingInterface Address**.
8. Turn on **VoIP** and select the **default** VoIP profile.
9. Select **OK**.

## Configuration steps - CLI

### To add the SIP proxy server firewall virtual IP and firewall address

1. Enter the following command to add the SIP proxy server firewall virtual IP.

```
config firewall vip
edit SIP_Proxy_VIP
set type static-nat
set extip 172.20.120.50
set mappedip 10.31.101.50
set extintf port1
end
```

2. Enter the following command to add the SIP proxy server firewall address.

```
config firewall address
edit SIP_Proxy_Server
set associated interface port2
set type ipmask
set subnet 10.31.101.50 255.255.255.255
end
```

### To add security policies

1. Enter the following command to add a destination NAT security policy that includes the SIP proxy server virtual IP that allows Phone B (and other SIP phones on the Internet) to send SIP request messages to the SIP proxy server.

```
config firewall policy
edit 0
set srcintf port1
set dstintf port2
set srcaddr all
set dstaddr SIP_Proxy_VIP
set action accept
set schedule always
set service SIP
set nat enable
set utm-status enable
set voip-profile default
end
```

2. Enter the following command to add a source NAT security policy to allow the SIP proxy server to send SIP request messages to Phone B and the Internet:

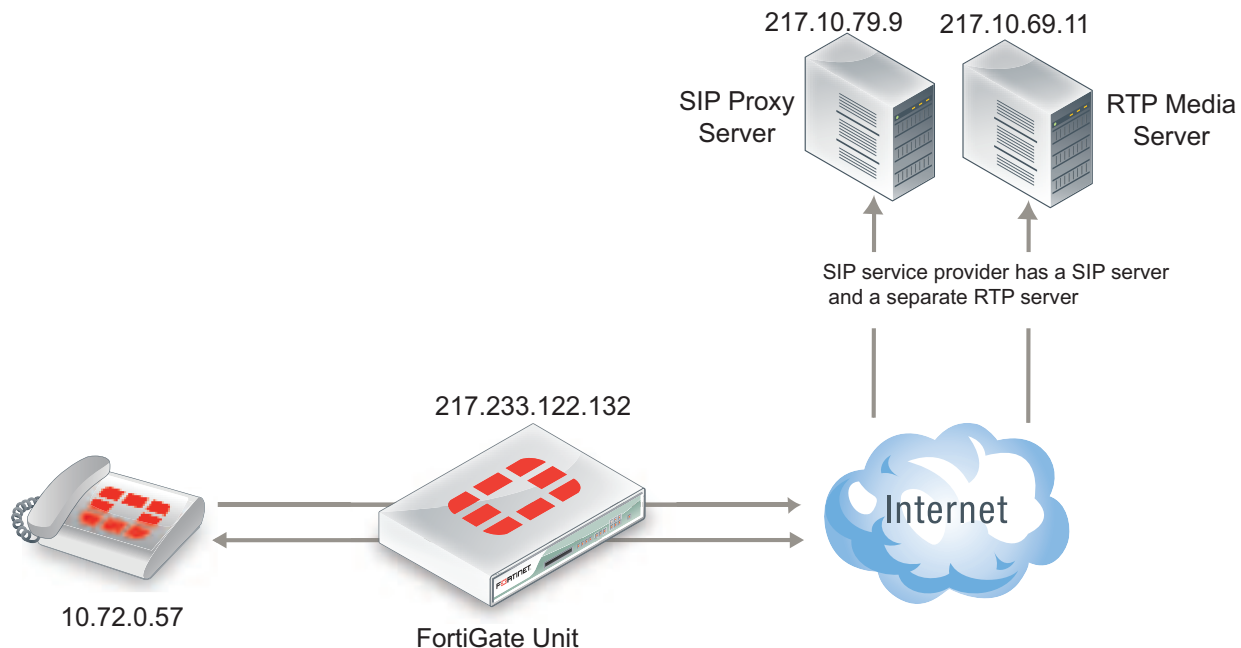
```
config firewall policy
edit 0
set srcintf port2
set dstintf port1
set srcaddr SIP_Proxy_Server
set dstaddr all
set action accept
set schedule always
set service SIP
set nat enable
set utm-status enable
set voip-profile default
end
```

## SIP and RTP source NAT

In the source NAT scenario shown below, a SIP phone connects to the Internet through a FortiGate with an IP address configured using PPPoE. The SIP ALG translates all private IPs in the SIP contact header into public IPs.

You need to configure an internal to external SIP security policy with NAT selected, and include a VoIP profile with SIP enabled.

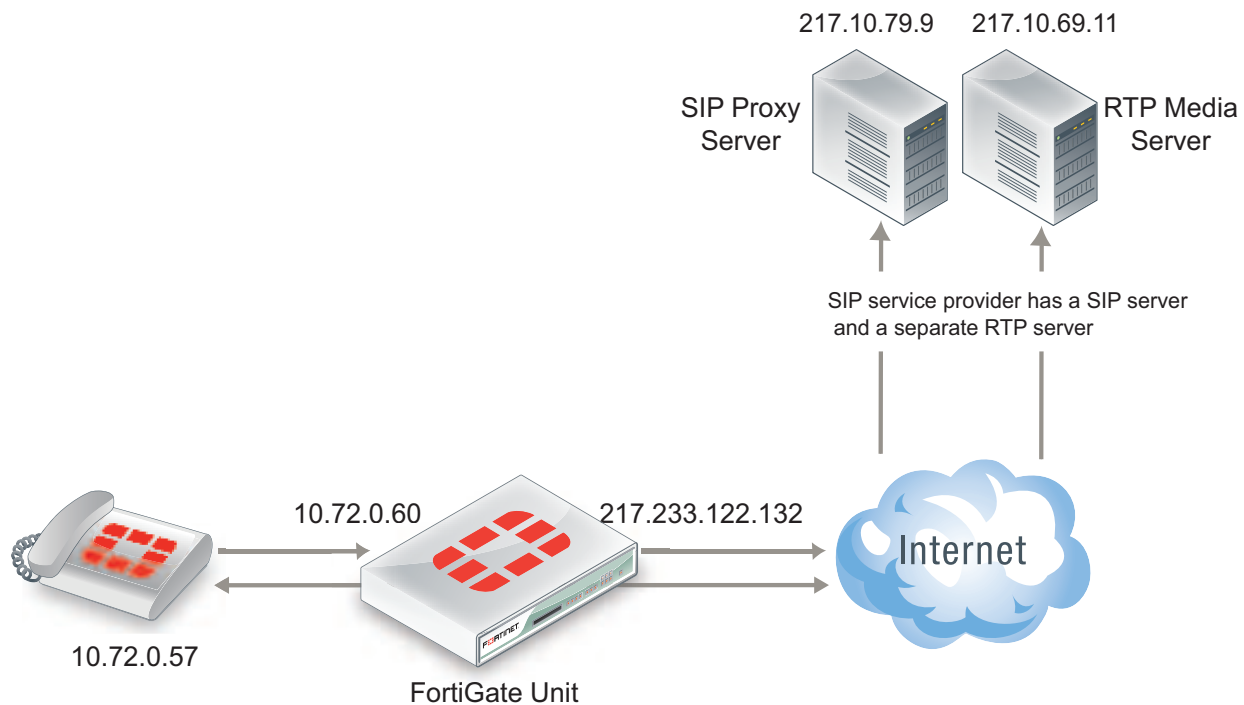
### SIP source NAT



## SIP and RTP destination NAT

In the following destination NAT scenario, a SIP phone can connect through the FortiGate to private IP address using a firewall virtual IP (VIP). The SIP ALG translates the SIP contact header to the IP of the real SIP proxy server located on the Internet.

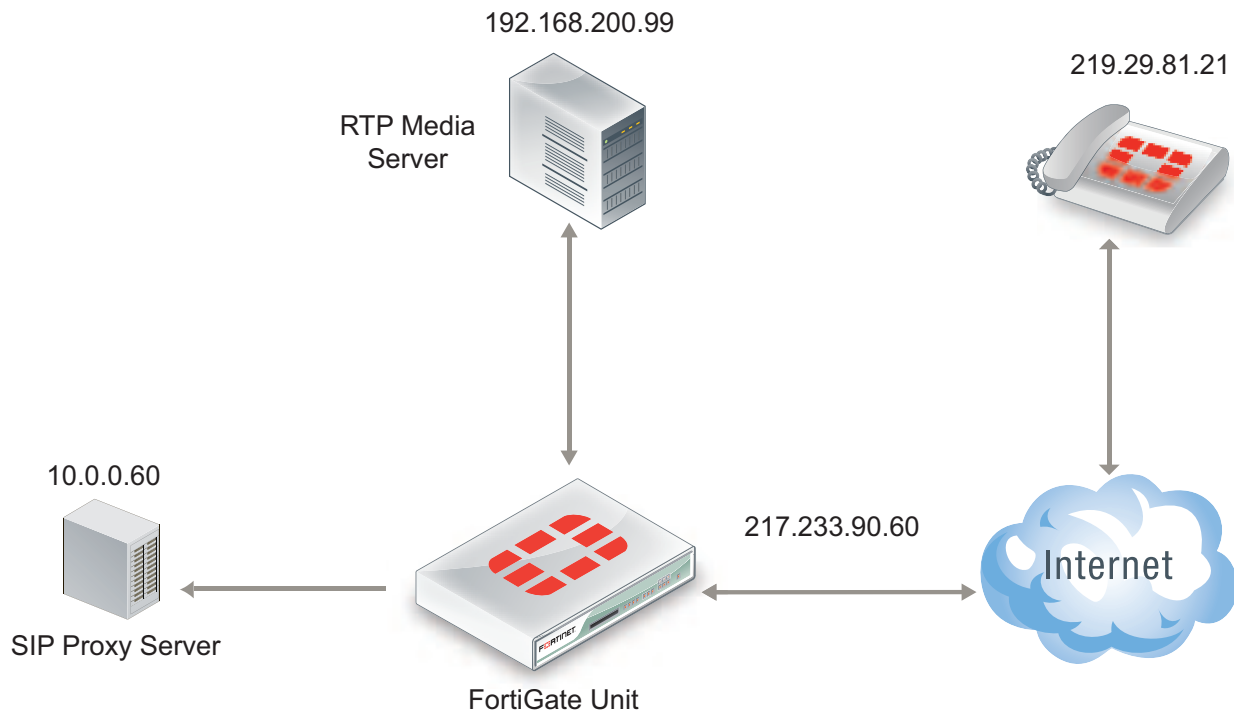
### SIP destination NAT



In the scenario, shown above, the SIP phone connects to a VIP (10.72.0.60). The SIP ALG translates the SIP contact header to 217.10.79.9, opens RTP pinholes, and manages NAT.

The FortiGate also supports a variation of this scenario where the RTP media server's IP address is hidden on a private network or DMZ.



**SIP destination NAT-RTP media server hidden**

In the scenario shown above, a SIP phone connects to the Internet. The VoIP service provider only publishes a single public IP. The FortiGate is configured with a firewall VIP. The SIP phone connects to the FortiGate (217.233.90.60) and using the VIP the FortiGate translates the SIP contact header to the SIP proxy server IP address (10.0.0.60). The SIP proxy server changes the SIP/SDP connection information (which tells the SIP phone which RTP media server IP it should contact) also to 217.233.90.60.

**Source NAT with an IP pool**

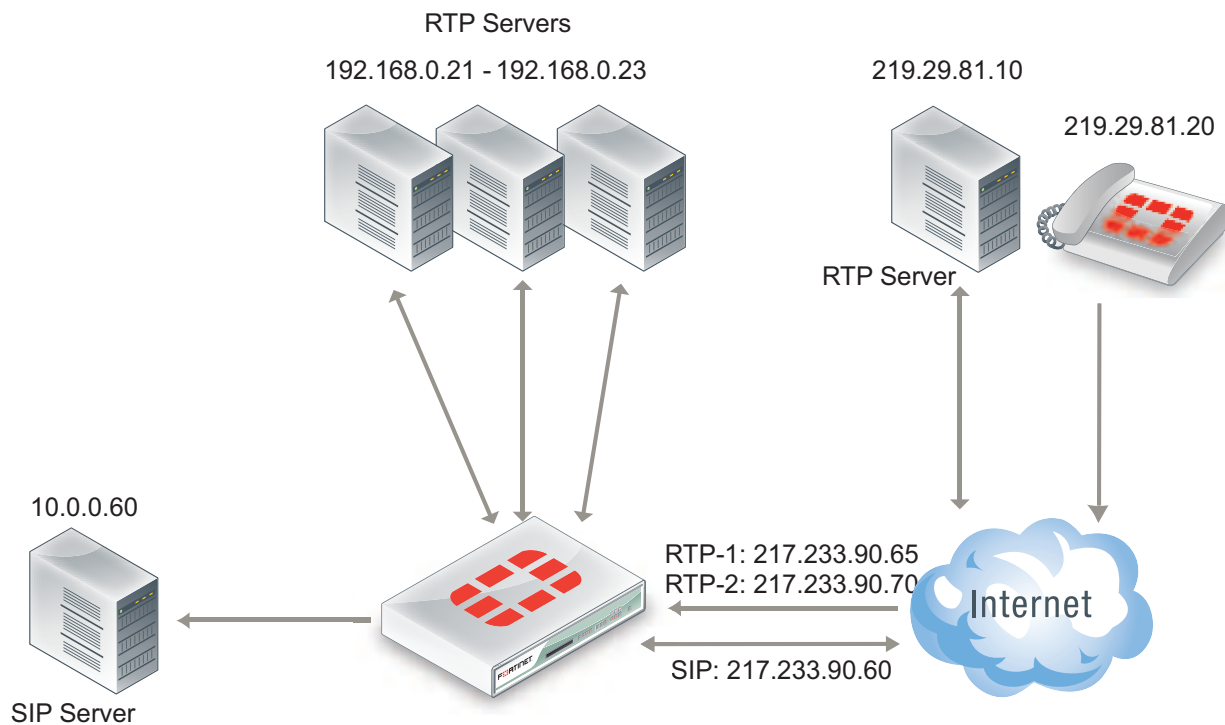
You can choose **NAT** with the **Dynamic IP Pool** option when configuring a security policy if the source IP of the SIP packets is different from the interface IP. The FortiGate ALG interprets this configuration and translates the SIP header accordingly.

This configuration also applies to destination NAT.

**Different source and destination NAT for SIP and RTP**

This is a more complex scenario that a SIP service provider may use. It can also be deployed in large-scale SIP environments where RTP has to be processed by the FortiGate and the RTP server IP has to be translated differently than the SIP server IP.

### Different source and destination NAT for SIP and RTP



In this scenario, shown above, assume there is a SIP server and a separate media gateway. The SIP server is configured so that the SIP phone (219.29.81.20) will connect to 217.233.90.60. The media gateway (RTP server: 219.29.81.10) will connect to 217.233.90.65.

What happens is as follows:

1. The SIP phone connects to the SIP VIP. The FortiGate ALG translates the SIP contact header to the SIP server: 219.29.81.20 > 217.233.90.60 (> 10.0.0.60).
2. The SIP server carries out RTP to 217.233.90.65.
3. The FortiGate ALG opens pinholes, assuming that it knows the ports to be opened.
4. RTP is sent to the RTP-VIP (217.233.90.65.) The FortiGate ALG translates the SIP contact header to 192.168.0.21.

## NAT with IP address conservation

In a source or destination NAT security policy that accepts SIP sessions, you can configure the SIP ALG or the SIP session helper to preserve the original source IP address of the SIP message in the `i=` line of the SDP profile. NAT with IP address conservation (also called SIP NAT tracing) changes the contents of SIP messages by adding the source IP address of the originator of the message into the SDP `i=` line of the SIP message. The SDP `i=` line is used for free-form text. However, if your SIP server can retrieve information from the SDP `i=` line, it can be useful for keeping a record of the source IP address of the originator of a SIP message when operating in a NAT environment. You can use this feature for billing purposes by extracting the IP address of the originator of the message.

## Configuring SIP IP address conservation for the SIP ALG

You can use the following command to enable or disable SIP IP address conservation in a VoIP profile for the SIP ALG. SIP IP address conservation is enabled by default in a VoIP profile.

```
config voip profile
  edit VoIP_Pro_1
    config sip
      set nat-trace disable
    end
  end
```

If the SIP message does not include an `i=` line and if the original source IP address of the traffic (before NAT) was 10.31.101.20 then the FortiGate would add the following `i=` line.

```
i=(o=IN IP4 10.31.101.20)
```

You can also use the `preserve-override` option to configure the SIP ALG to either add the original `o=` line to the end of the `i=` line or replace the `i=` line in the original message with a new `i=` line in the same form as above for adding a new `i=` line.

By default, `preserve-override` is disabled and the SIP ALG adds the original `o=` line to the end of the original `i=` line. Use the following command to configure the SIP ALG to replace the original `i=` line:

```
config voip profile
  edit VoIP_Pro_1
    config sip
      set preserve-override enable
    end
  end
```

## Configuring SIP IP address conservation for the SIP session helper

You can use the following command to enable or disable SIP IP address conservation for the SIP session helper. IP address conservation is enabled by default for the SIP session helper.

```
config system settings
  set sip-nat-trace disable
end
```

If the SIP message does not include an `i=` line and if the original source IP address of the traffic (before NAT) was 10.31.101.20 then the FortiGate would add the following `i=` line.

```
i=(o=IN IP4 10.31.101.20)
```

## Controlling how the SIP ALG NATs SIP contact header line addresses

You can enable `contact-fixup` so that the SIP ALG performs normal SIP NAT translation to SIP contact headers as SIP messages pass through the FortiGate.

Disable `contact-fixup` if you do not want the SIP ALG to perform normal NAT translation of the SIP contact header if a Record-Route header is also available. If `contact-fixup` is disabled, the FortiGate ALG does the following with contact headers:

- For Contact in Requests, if a Record-Route header is present and the request comes from the external network, the SIP Contact header is not translated.

- For Contact in Responses, if a Record-Route header is present and the response comes from the external network, the SIP Contact header is not translated.

If `contact-fixup` is disabled, the SIP ALG must be able to identify the external network. To identify the external network, you must use the `config system interface` command to set the `external` keyword to `enable` for the interface that is connected to the external network.

Enter the following command to perform normal NAT translation of the SIP contact header:

```
config voip profile
  edit VoIP_Pro_1
    config sip
      set contact-fixup enable
    end
  end
```

## Controlling NAT for addresses in SDP lines

You can use the `no-sdp-fixup` option to control whether the FortiGate performs NAT on addresses in SDP lines in the SIP message body.

The `no-sdp-fixup` option is disabled by default and the FortiGate performs NAT on addresses in SDP lines. Enable this option if you don't want the FortiGate to perform NAT on the addresses in SDP lines.

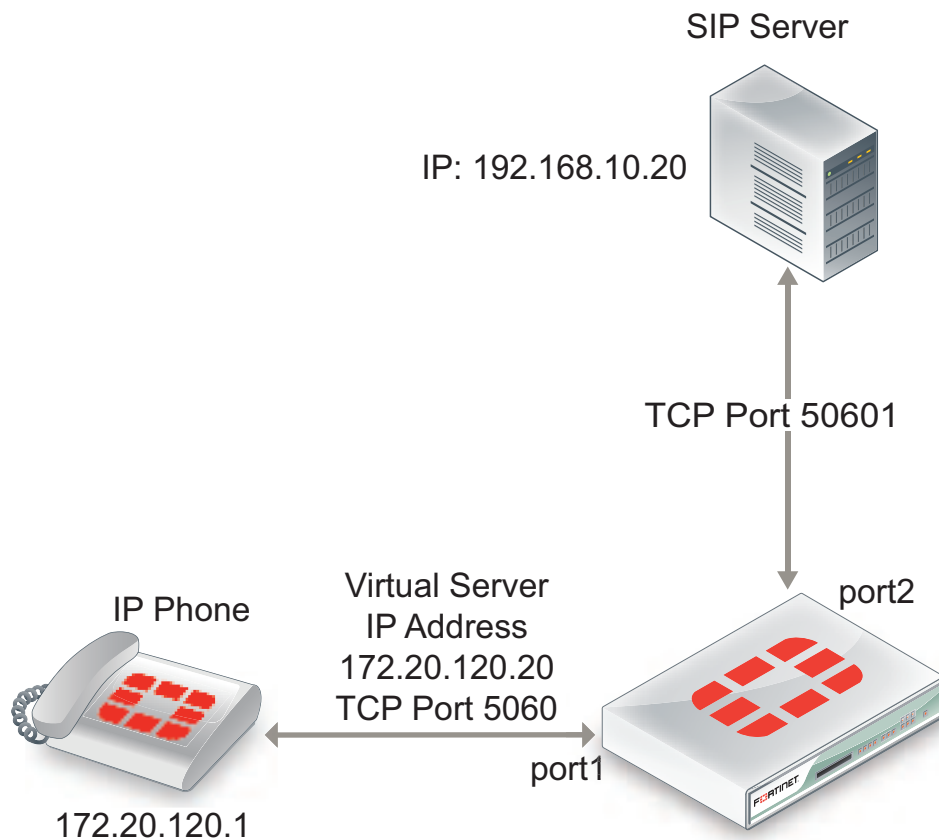
```
config voip profile
  edit VoIP_Pro_1
    config sip
      set no-sdp-fixup enable
    end
  end
```

## Translating SIP session destination ports

Using port forwarding virtual IPs you can change the destination port of SIP sessions as they pass through the FortiGate.

### Translating SIP sessions to a different destination port

To configure translating SIP sessions to a different destination port you must add a static NAT virtual IP that translates the SIP destination port to another port destination. In the example the destination port is translated from 5060 to 50601. This configuration can be used if SIP sessions use different destination ports on different networks.

**Example translating SIP sessions to a different destination port****To translate SIP sessions to a different destination port**

1. Add the static NAT virtual IP.

This virtual IP forwards traffic received at the port1 interface for IP address 172.20.120.20 and destination port 5060 to the SIP server at IP address 192.168.10.20 with destination port 5061.

```
config firewall vip
  edit "sip_port_trans_vip"
    set type static-nat
    set portforward enable
    set protocol tcp
    set extip 172.20.120.20
    set extport 5060
    set extintf "port1"
    set mappedip 192.168.10.20
    set mappedport 50601
    set comment "Translate SIP destination port"
  end
```

2. Add a security policy that includes the virtual IP and the default VoIP profile.

```
config firewall policy
  edit 1
    set srcintf "port1"
    set dstintf "port2"
    set srcaddr "all"
```

```

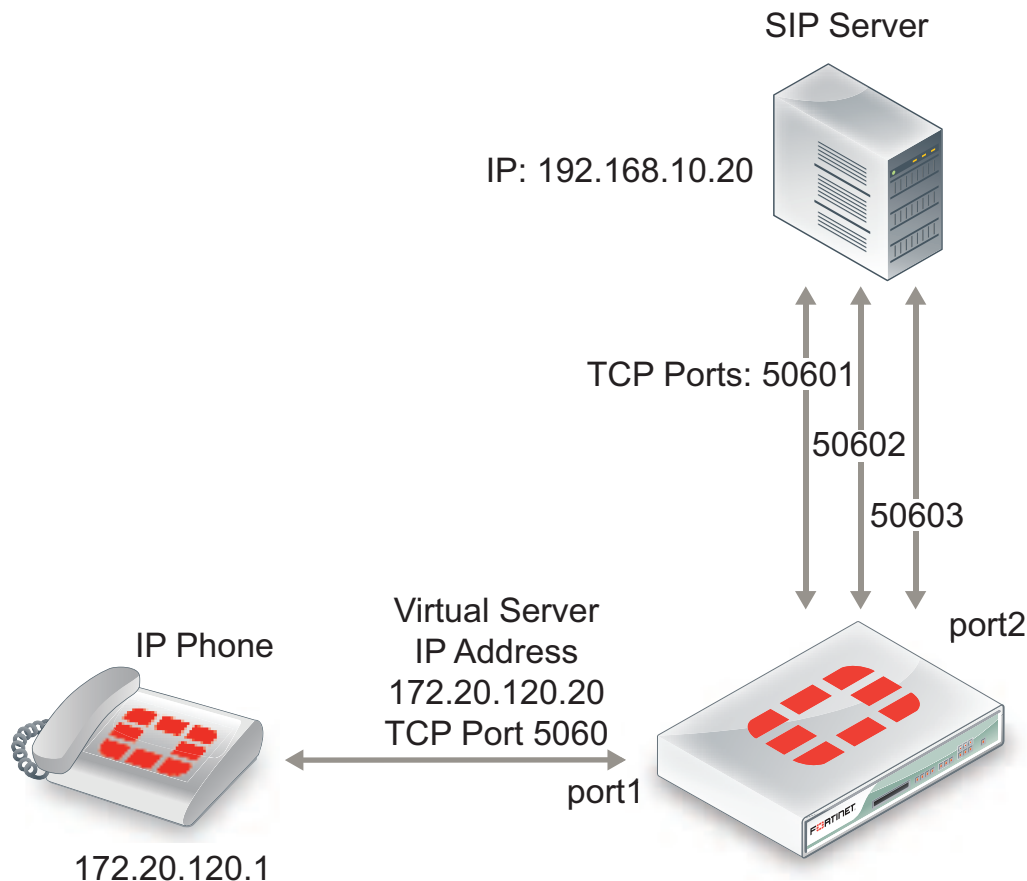
set dstaddr "sip_port_trans_vip"
set action accept
set schedule "always"
set service "ANY"
set utm-status enable
set profile-protocol-options default
set comments "Translate SIP destination port"
end

```

## Translating SIP sessions to multiple destination ports

You can use a load balance virtual IP to translate SIP session destination ports to a range of destination ports. In this example the destination port is translated from 5060 to the range 50601 to 50603. This configuration can be used if your SIP server is configured to receive SIP traffic on multiple ports.

### Example translating SIP traffic to multiple destination ports



### To translated SIP sessions to multiple destination ports

1. Add the load balance virtual IP.

This virtual IP forwards traffic received at the port1 interface for IP address 172.20.120.20 and destination port 5060 to the SIP server at IP address 192.168.10.20 with destination port 5061.

```
config firewall vip
edit "sip_port_ldbl_vip"
set type load-balance
set portforward enable
set protocol tcp
set extip 172.20.120.20
set extport 5060
set extintf "port1"
set mappedip 192.168.10.20
set mappedport 50601-50603
set comment "Translate SIP destination port range"
end
```

## 2. Add a security policy that includes the virtual IP and VoIP profile.

```
config firewall policy
edit 1
set srcintf "port1"
set dstintf "port2"
set srcaddr "all"
set dstaddr "sip_port_ldbl_vip"
set action accept
set schedule "always"
set service "ANY"
set utm-status enable
set voip-profile default
set comments "Translate SIP destination port"
end
```

## Adding the original IP address and port to the SIP message header after NAT

In some cases your SIP configuration may require that the original IP address and port from the SIP contact request is kept after NAT. For example, the original SIP contact request could include the following:

```
Contact: <sip:0150302438@172.20.120.110:5060>;
```

After the packet goes through the FortiGate and NAT is performed, the contact request could normally look like the following (the IP address translated to a different IP address and the port to a different port):

```
Contact: <sip:0150302438@10.10.10.21:33608>;
```

You can enable `register-contact-trace` in a VoIP profile to have the SIP ALG add the original IP address and port in the following format:

```
Contact: <sip:0150302438@<nated-ip>:<nated-port>;o=<original-ip>: <original-port>>;
```

So the contact line after NAT could look like the following:

```
Contact: <sip:0150302438@10.10.10.21:33608;o=172.20.120.110:5060>;
```

Enter the following command to enable keeping the original IP address and port:

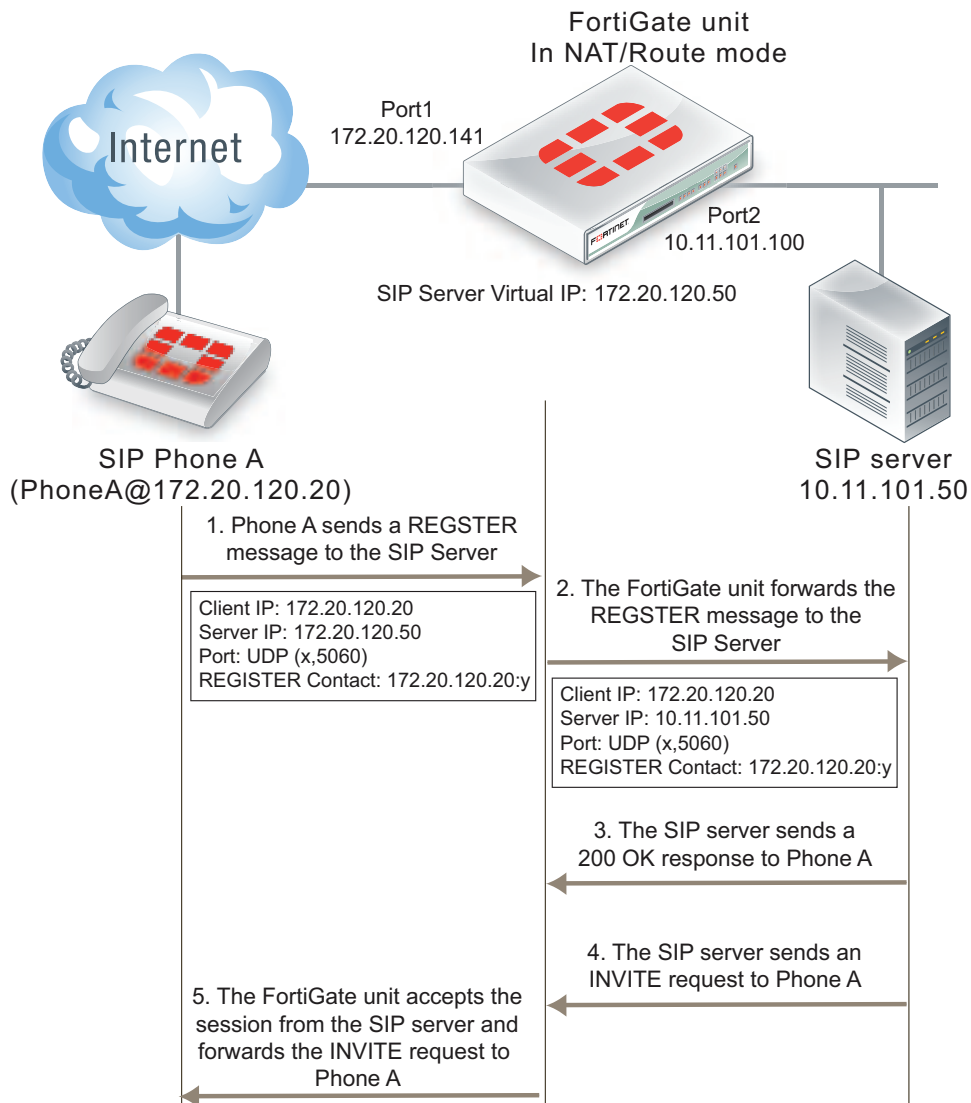
```
config voip profile
edit Profile_name
config sip
set register-contract-trace enable
end
```

## Enhancing SIP pinhole security

You can use the `strict-register` option in a SIP VoIP profile to open smaller pinholes. This option is enabled by default on the default VoIP profiles and in all new VoIP profiles that you create.

As shown below, when FortiGate is protecting a SIP server on a private network, the FortiGate does not have to open a pinhole for the SIP server to send INVITE requests to a SIP Phone on the Internet after the SIP Phone has registered with the server.

### FortiGate protecting a SIP server on a private network



In the example, a client (SIP Phone A) sends a REGISTER request to the SIP server with the following information:

```
Client IP: 10.31.101.20
Server IP: 10.21.101.50
```



```
Port: UDP (x,5060)
REGISTER Contact: 10.31.101.20:y
```

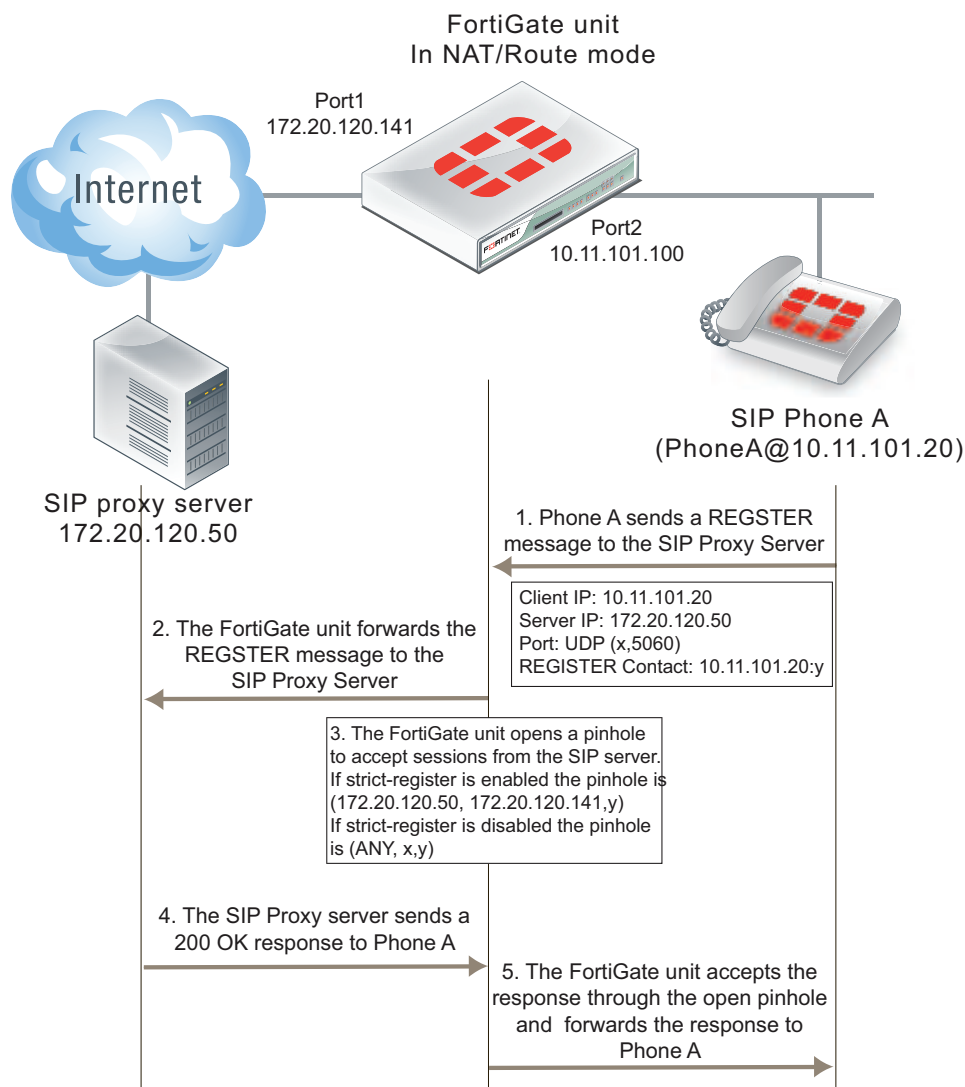
Where  $x$  and  $y$  are ports chosen by Phone A.

As soon as the server sends the 200 OK reply it can forward INVITE requests from other SIP phones to SIP Phone A. If the SIP proxy server uses the information in the REGISTER message received from SIP Phone A the INVITE messages sent to Phone A will only get through the FortiGate if a policy has been added to allow the server to send traffic from the private network to the Internet. Or the SIP ALG must open a pinhole to allow traffic from the server to the Internet. In most cases the FortiGate is protecting the SIP server so there is no reason not to add a security policy to allow the SIP server to send outbound traffic to the Internet.

In a typical SOHO scenario, shown below, SIP Phone A is being protected from the Internet by a FortiGate. In most cases the FortiGate would not allow incoming traffic from the Internet to reach the private network. So the only way that an INVITE request from the SIP server can reach SIP Phone A is if the SIP ALG creates an incoming pinhole. All pinholes have three attributes:

```
(source address, destination address, destination port)
```

### SOHO configuration, FortiGate protecting a network with SIP phones



The more specific a pinhole is the more secure it is because it accept less traffic. In this situation, the pinhole would be more secure if it only accepted traffic from the SIP server. This is what happens if `strict-register` is enabled in the VoIP profile that accepts the REGISTER request from Phone A.

(SIP server IP address, client IP address, destination port)

If `strict-register` is disabled (the default configuration) the pinhole is set up with the following attributes

(ANY IP address, client IP address, destination port)

This pinhole allows connections through the FortiGate from ANY source address which is a much bigger and less secure pinhole. In most similar network configurations you should enable `strict-register` to improve pinhole security.

Enabling `strict-register` can cause problems when the SIP registrar and SIP proxy server are separate entities with separate IP addresses.

Enter the following command to enable `strict-register` in a VoIP profile.

```
config voip profile
  edit Profile_name
    config sip
      set strict-register enable
    end
```

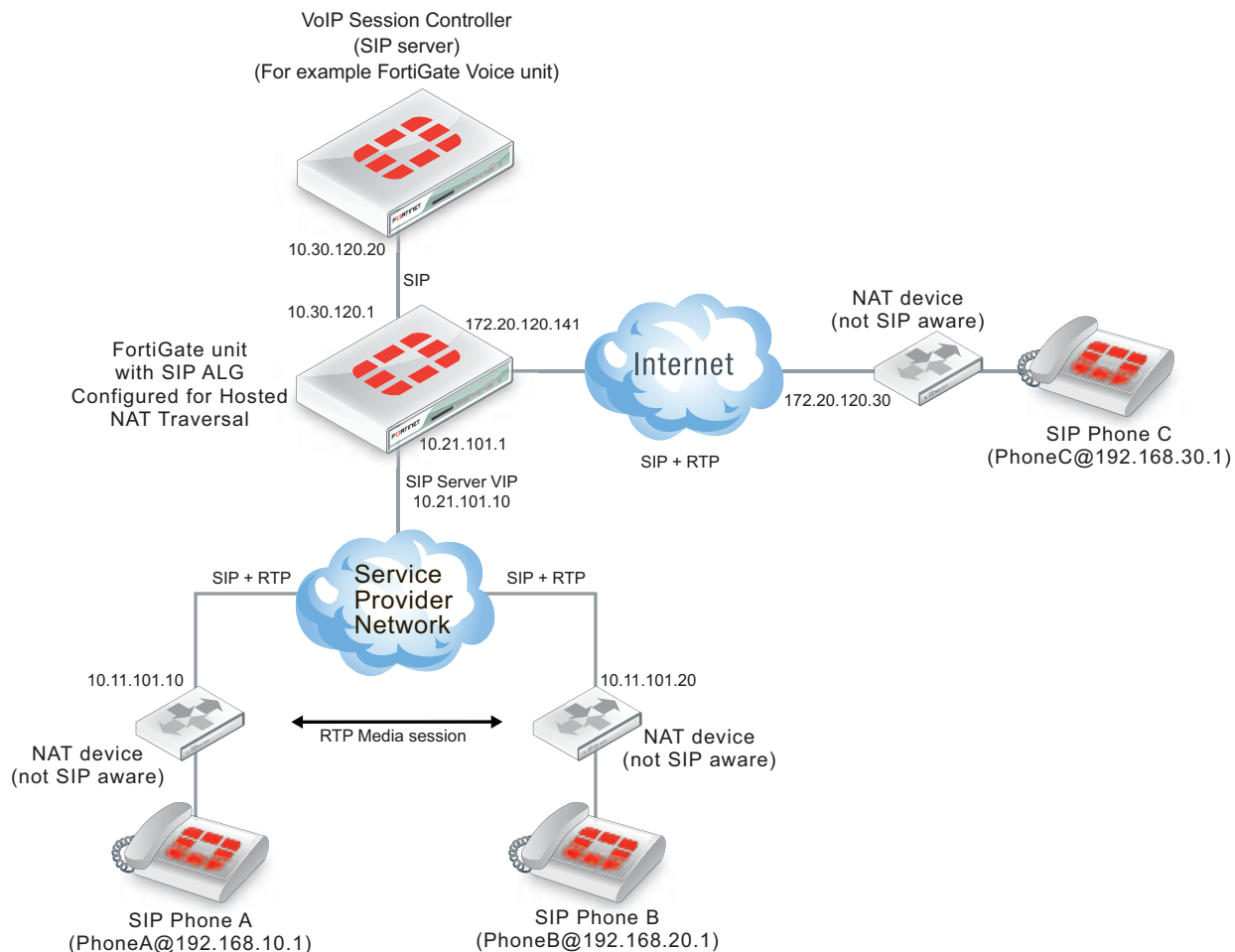
## Hosted NAT traversal

With the increase in the use of VoIP and other media traffic over the Internet, service provider network administrators must defend their networks from threats while allowing voice and multimedia traffic to flow transparently between users and servers and among users. A common scenario could involve providing SIP VoIP services for customers with SIP phones installed behind NAT devices that are not SIP aware. NAT devices that are not SIP aware cannot translate IP addresses in SIP headers and SDP lines in SIP packets but can and do perform source NAT on the source or addresses of the packets. In this scenario the user's SIP phones would communicate with a SIP proxy server to set up calls between SIP phones. Once the calls are set up RTP packets would be communicated directly between the phones through each user's NAT device.

The problem with this configuration is that the SIP headers and SDP lines in the SIP packets sent from the phones and received by the SIP proxy server would contain the private network addresses of the VoIP phones that would not be routable on the service provider network or on the Internet. One solution could be to for each customer to install and configure SIP aware NAT devices. If this is not possible, another solution requires implement hosted NAT traversal.

In a hosted NAT traversal (HNT) configuration, a FortiGate is installed between the NAT device and the SIP proxy server and configured with a VoIP profile that enables SIP hosted NAT traversal. Security policies that include the VoIP profile also support destination NAT using a firewall virtual IP. When the SIP phones connect to the SIP server IP address the security policy accepts the SIP packets, the virtual IP translates the destination addresses of the packets to the SIP server IP address, and the SIP ALG NAT traversal configuration translates the source IP addresses on the SIP headers and SDP lines to the source address of the SIP packets (which would be the external IP address of the NAT devices). The SIP server then sees the SIP phone IP address as the external IP address of the NAT device. As a result SIP and RTP media sessions are established using the external IP addresses of the NAT devices instead of the actual IP addresses of the SIP phones.

## FortiGate SIP Hosted NAT Traversal configuration



## Configuration example: Hosted NAT traversal for calls between SIP Phone A and SIP Phone B

The following address translation takes place to allow a SIP call from SIP Phone A to SIP Phone B in the above diagram.

1. SIP Phone A sends a SIP Invite message to the SIP server. Packet source IP address: 192.168.10.1, destination IP address: 10.21.101.10.
2. The SIP packets are received by the NAT device which translates the source address of the SIP packets from 192.168.10.1 to 10.11.101.20.
3. The SIP packets are received by the FortiGate which translates the packet destination IP address to 10.30.120.20. The SIP ALG also translates the IP address of the SIP phone in the SIP header and SDP lines from 192.168.10.1 to 10.11.101.20.
4. The SIP server accepts the Invite message and forwards it to SIP Phone B at IP address 10.11.101.20. The SIP server has this address for SIP Phone B because SIP packets from SIP Phone B have also been translated using the hosted NAT traversal configuration of the SIP ALG.

5. When the SIP call is established, the RTP session is between 10.11.101.10 and 10.11.101.20 and does not pass through the FortiGate. The NAT devices translated the destination address of the RTP packets to the private IP addresses of the SIP phones.

## General configuration steps

The following general configuration steps are required for this destination NAT SIP configuration. This example uses the default VoIP profile.

1. Add a VoIP profile that enables hosted NAT translation.
2. Add a SIP proxy server firewall virtual IP.
3. Add a firewall address for the SIP proxy server on the private network.
4. Add a destination NAT security policy that accepts SIP sessions from the Internet destined for the SIP proxy server virtual IP and translates the destination address to the IP address of the SIP proxy server on the private network.
5. Add a security policy that accepts SIP sessions initiated by the SIP proxy server and destined for the Internet.

## Configuration steps - GUI

### To add the SIP proxy server firewall virtual IP

1. Go to **Policy & Objects > Virtual IPs**.
2. Add the SIP proxy server virtual IP.

|                                  |               |
|----------------------------------|---------------|
| <b>Name</b>                      | SIP_Proxy_VIP |
| <b>External Interface</b>        | port1         |
| <b>Type</b>                      | Static NAT    |
| <b>External IP Address/Range</b> | 172.20.120.50 |
| <b>Mapped IP Address/Range</b>   | 10.31.101.50  |

### To add a firewall address for the SIP proxy server

1. Go to **Policy & Objects > Addresses**.
2. Add the following for the SIP proxy server:

|                          |                              |
|--------------------------|------------------------------|
| <b>Category</b>          | Address                      |
| <b>Name</b>              | SIP_Proxy_Server             |
| <b>Type</b>              | Subnet                       |
| <b>Subnet / IP Range</b> | 10.31.101.50/255.255.255.255 |
| <b>Interface</b>         | port2                        |

**To add the security policies**

1. Go to **Policy & Objects > IPv4 Policy**.
2. Add a destination NAT security policy that includes the SIP proxy server virtual IP that allows Phone B (and other SIP phones on the Internet) to send SIP request messages to the SIP proxy server.

|                            |               |
|----------------------------|---------------|
| <b>Incoming Interface</b>  | port1         |
| <b>Outgoing Interface</b>  | port2         |
| <b>Source</b>              | all           |
| <b>Destination Address</b> | SIP_Proxy_VIP |
| <b>Schedule</b>            | always        |
| <b>Service</b>             | SIP           |
| <b>Action</b>              | ACCEPT        |

3. Turn on **NAT** and select **Use Outgoing Interface Address**.
4. Turn on **VoIP** and select the **default** VoIP profile.
5. Select **OK**.
6. Add a source NAT security policy to allow the SIP proxy server to send SIP request messages to Phone B and the Internet:

|                            |                  |
|----------------------------|------------------|
| <b>Incoming Interface</b>  | port2            |
| <b>Outgoing Interface</b>  | port1            |
| <b>Source</b>              | SIP_Proxy_Server |
| <b>Destination Address</b> | all              |
| <b>Schedule</b>            | always           |
| <b>Service</b>             | SIP              |
| <b>Action</b>              | ACCEPT           |

7. Turn on **NAT** and select **Use Outgoing Interface Address**.
8. Turn on **VoIP** and select the **default** VoIP profile.
9. Select **OK**.

## Configuration steps - CLI

**To add a VoIP profile that enables hosted NAT translation**

1. Enter the following command to add a VoIP profile named HNT that enables hosted NAT traversal. This command shows how to clone the default VoIP profile and enable hosted NAT traversal.

```
config voip profile
clone default to HNT
```

```
edit HNT
  config sip
    set hosted-nat-traversal enable
  end
end
```

### To add the SIP proxy server firewall virtual IP and firewall address

1. Enter the following command to add the SIP proxy server firewall virtual IP.

```
config firewall vip
  edit SIP_Proxy_VIP
    set type static-nat
    set extip 10.21.101.10
    set mappedip 10.30.120.20
    set extintf port1
  end
```

2. Enter the following command to add the SIP proxy server firewall address.

```
config firewall address
  edit SIP_Proxy_Server
    set associated interface port2
    set type ipmask
    set subnet 10.30.120.20 255.255.255.255
  end
```

### To add security policies

1. Enter the following command to add a destination NAT security policy that includes the SIP proxy server virtual IP that allows Phone A to send SIP request messages to the SIP proxy server.

```
config firewall policy
  edit 0
    set srcintf port1
    set dstintf port2
    set srcaddr all
    set dstaddr SIP_Proxy_VIP
    set action accept
    set schedule always
    set service SIP
    set nat enable
    set utm-status enable
    set profile-protocol-options default
    set voip-profile HNT
  end
```

2. Enter the following command to add a source NAT security policy to allow the SIP proxy server to send SIP request messages to Phone B:

```
config firewall policy
  edit 0
    set srcintf port2
    set dstintf port1
    set srcaddr SIP_Proxy_Server
    set dstaddr all
    set action accept
    set schedule always
    set service SIP
    set nat enable
```

```
set utm-status enable
set profile-protocol-options default
set voip-profile default
end
```

## Hosted NAT traversal for calls between SIP Phone A and SIP Phone C

The following address translation takes place to allow a SIP call from SIP Phone A to SIP Phone C in the previous diagram.

1. SIP Phone A sends a SIP Invite message to the SIP server. Packet source IP address: 192.168.10.1 and destination IP address: 10.21.101.10.
2. The SIP packets are received by the NAT device which translates the source address of the SIP packets from 192.168.10.1 to 10.11.101.20.
3. The SIP packets are received by the FortiGate which translates the packet destination IP address to 10.30.120.20. The SIP ALG also translates the IP address of the SIP phone in the SIP header and SDP lines from 192.168.10.1 to 10.11.101.20.
4. The SIP server accepts the Invite message and forwards it to SIP Phone C at IP address 172.20.120.30. The SIP server has this address for SIP Phone C because SIP packets from SIP Phone C have also been translated using the hosted NAT traversal configuration of the SIP ALG.
5. When the SIP call is established, the RTP session is between 10.11.101.10 and 172.20.120.30. The packets pass through the FortiGate which performs NAT as required.

## Restricting the RTP source IP

Use the following command in a VoIP profile to restrict the RTP source IP to be the same as the SIP source IP when hosted NAT traversal is enabled.

```
config voip profile
edit VoIP_HNT
config sip
set hosted-nat-traversal enable
set hnt-restrict-source-ip enable
end
end
```



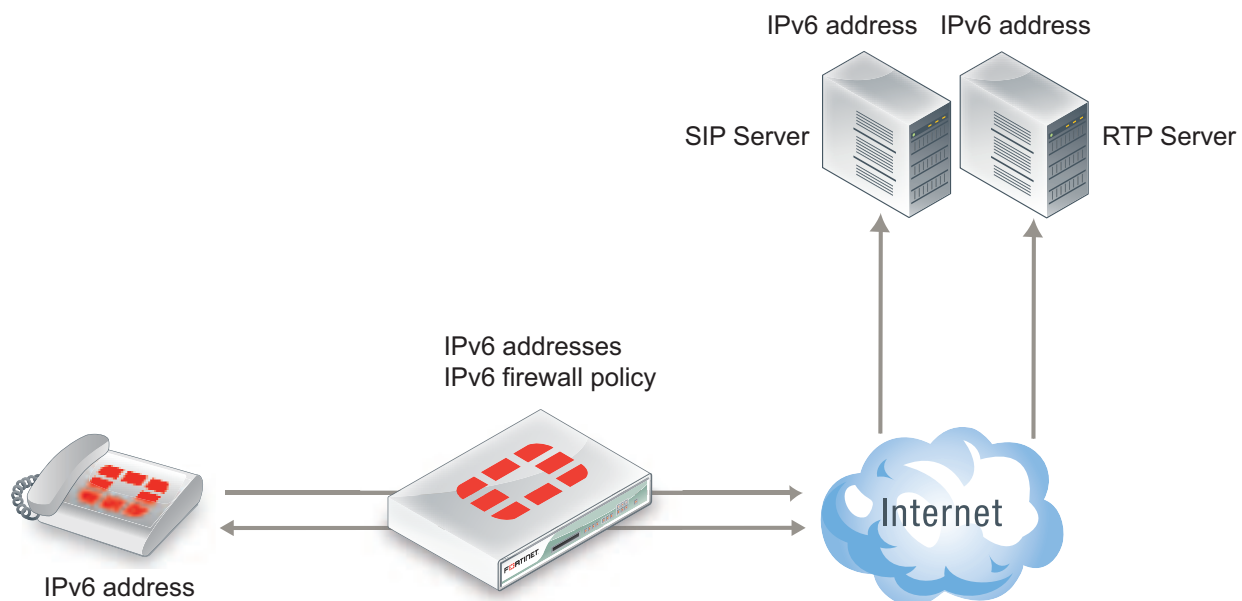
## SIP over IPv6

FortiGate operating in NAT/Route and in Transparent mode support SIP over IPv6. The SIP ALG can process SIP messages that use IPv6 addresses in the headers, bodies, and in the transport stack. The SIP ALG cannot modify the IPv6 addresses in the SIP headers so FortiGate cannot perform SIP or RTP NAT over IPv6 and also cannot translate between IPv6 and IPv4 addresses.

In the scenario shown in the figure below, a SIP phone connects to the Internet through a FortiGate operating. The phone and the SIP and RTP servers all have IPv6 addresses.

The FortiGate has IPv6 security policies that accept SIP sessions. The SIP ALG understands IPv6 addresses and can forward IPv6 sessions to their destinations. Using SIP application control features the SIP ALG can also apply rate limiting and other settings to SIP sessions.

### SIP support for IPv6

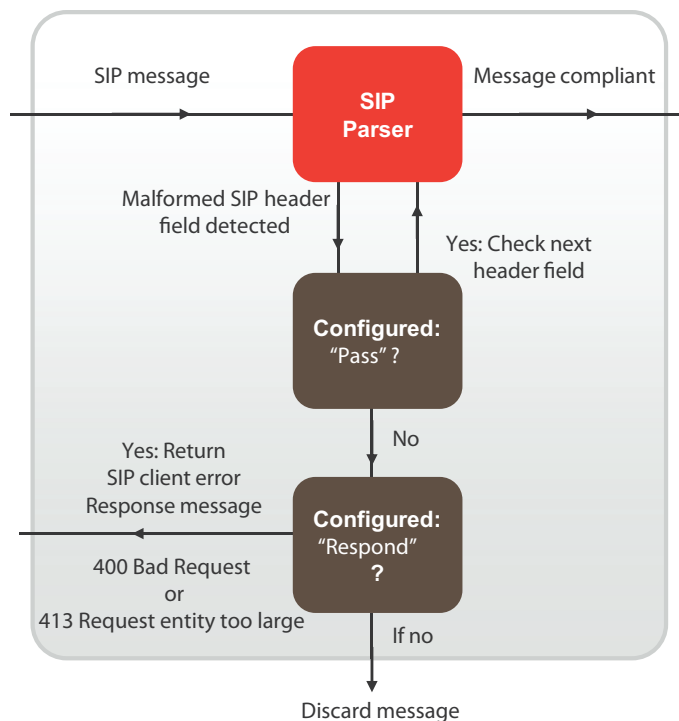


To enable SIP support for IPv6 add an IPv6 security policy that accepts SIP packets and includes a VoIP profile.

# Deep SIP message inspection

Deep SIP message syntax inspection (also called Deep SIP header inspection or SIP fuzzing protection) provides protection against malicious SIP messages by applying SIP header and SDP profile syntax checking. SIP Fuzzing attacks can be used by attackers to discover and exploit vulnerabilities of a SIP entity (for example a SIP proxy server). Most often these attacks could crash or compromise the SIP entity.

## Deep SIP message inspection



- Checks the SIP request message Request-line
- Checks the following SIP header fields:
  - Allow, Call-id, Contact, Content-length, Content-type, CSeq, Expires, From, Max-Forwards, P-asserted-identity, Rack, Record-Route, Route, Rseq, To, Via
- Checks all SDP profile lines
- Configurable header and body length checks
- Optional logging of message violations

Deep SIP message inspection checks the syntax of each SIP header and SDP profile line to make sure they conform to the syntax defined in the relevant RFC and IETF standard. You can also configure the SIP ALG to inspect for:

- Unknown SIP message types (message types not defined in a SIP RFC) this option is enabled by default and can be disabled. When enabled unknown message types are discarded. Configured using the `block-unknown` option.
- Unknown line types (message line types that are not defined in any SIP or SDP RFC). Configured using the `unknown-header` option.
- Messages that are longer than a configured maximum size. Configured using the `max-body-length` option.
- Messages that contain one or more lines that are longer than a set maximum line length (default 998 characters). Configured using the `max-line-length` option.

## Actions taken when a malformed message line is found

When a malformed message line or other error is found the SIP ALG can be configured to discard the message containing the error, pass the message without any other actions, or responding to the message with a 400 Bad Request or 413 Request entity too large client error SIP response message and then discard the message. (For information about client error SIP response messages, see [Client error on page 28](#).)

If a message line is longer than the configured maximum, the SIP ALG sends the following message:

```
SIP/2.0 413 Request Entity Too Large, <optional_info>
```

If a message line is incorrect or in an unknown message line is found, the SIP ALG sends the following message:

```
SIP/2.0 400 Bad Request, <optional_info>
```

The `<optional_info>` provides more information about why the message was rejected. For example, if the SIP ALG finds a malformed Via header line, the response message may be:

```
SIP/2.0 400 Bad Request, malformed Via header
```

If the SIP ALG finds a malformed message line, and the action for this message line type is discard, the message is discarded with no further checking or responses. If the action is pass, the SIP ALG continues parsing the SIP message for more malformed message lines. If the action is respond, the SIP ALG sends the SIP response message and discards the message containing the malformed line with no further checking or response. If only malformed message line types with action set to pass are found, the SIP ALG extracts as much information as possible from the message (for example for NAT and opening pinholes, and forwards the message to its destination).

If a SIP message containing a malformed line is discarded the SIP ALG will not use the information in the message for call processing. This could result in the call being terminated. If a malformed line in a SIP message includes information required for the SIP call that the SIP ALG cannot interpret (for example, if an IP address required for SIP NAT is corrupted) the SIP ALG may not be able to continue processing the call and it could be terminated. Discarded messages are counted by SIP ALG static message counters.

## Logging and statistics

To record a log message each time the SIP ALG finds a malformed header, enable logging SIP violations in a VoIP profile. In all cases, when the SIP ALG finds an error the FortiGate records a malformed header log message that contains information about the error. This happens even if the action is set to pass.

If, because of recording log messages for deep message inspection, the CPU performance is affected by a certain amount, the FortiGate records a critical log message about this event and stops writing log messages for deep SIP message inspection.

The following information is recorded in malformed header messages:

- The type of message line in which the error was found.
- The content of the message line in which the error was found (it will be truncated if it makes the log message too long)
- The column or character number in which the error was found (to make it easier to determine what caused the error)

## Deep SIP message inspection best practices

Because of the risks imposed by SIP header attacks or incorrect data being allowed and because selecting drop or respond does not require more CPU overhead than pass you would want to set all tests to drop or respond. However, in some cases malformed lines may be less of a threat or risk. For example, the SDP `i=` does not usually contain information that is parsed by any SIP device so a malformed `i=` line may not pose a threat.

You can also use the pre-defined VoIP profiles to apply different levels of deep message inspection. The default VoIP profile sets all deep message inspection options to pass and the strict VoIP profile sets all deep message inspection options to discard. From the CLI you can use the `clone` command to copy these pre-defined VoIP profiles and then customize them for your requirements.

## Configuring deep SIP message inspection

You configure deep SIP message inspection in a VoIP profile. All deep SIP message inspection options are available only from the CLI.

Enter the following command to configure deep SIP message inspection to discard messages with malformed Request-lines (the first line in a SIP request message):

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set malformed-request-line respond
    end
  end
```



You cannot configure message inspection for the Status-line, which is the first line in a SIP response message.

The following table lists the SIP header lines that the SIP ALG can inspect and the CLI command for configuring the action for each line type. The table also lists the RFC that the header line is defined in.

### SIP header lines that the SIP ALG can inspect for syntax errors

| SIP Header line       | VoIP profile option             | RFC      |
|-----------------------|---------------------------------|----------|
| <b>Allow</b>          | malformed-header-allow          | RFC 3261 |
| <b>Call-ID</b>        | malformed-header-call-id        | RFC 3261 |
| <b>Contact</b>        | malformed-header-contact        | RFC 3261 |
| <b>Content-Length</b> | malformed-header-content-length | RFC 3261 |
| <b>Content-Type</b>   | malformed-header-content-type   | RFC 3261 |

| SIP Header line            | VoIP profile option                  | RFC      |
|----------------------------|--------------------------------------|----------|
| <b>CSeq</b>                | malformed-header-cseq                | RFC 3261 |
| <b>Expires</b>             | malformed-header-expires             | RFC 3261 |
| <b>From</b>                | malformed-header-from                | RFC 3261 |
| <b>Max-forwards</b>        | malformed-header-max-forwards        | RFC 3261 |
| <b>P-Asserted-Identity</b> | malformed-header-p-asserted-identity | RFC 3325 |
| <b>RAck</b>                | malformed-header-rack                | RFC 3262 |
| <b>Record-Route</b>        | malformed-header-record-route        | RFC 3261 |
| <b>Route</b>               | malformed-header-route               | RFC 3261 |
| <b>RSeq</b>                | malformed-header-rseq                | RFC 3262 |
| <b>To</b>                  | malformed-header-to                  | RFC 3261 |
| <b>Via</b>                 | malformed-header-via                 | RFC 3261 |

The table below lists the SDP profile lines that the SIP ALG inspects and the CLI command for configuring the action for each line type. SDP profile lines are defined by RFC 4566 and RFC 2327.

#### SDP profile lines that the SIP ALG can inspect for syntax errors

| Attribute | VoIP profile option    |
|-----------|------------------------|
| <b>a=</b> | malformed-header-sdb-a |
| <b>b=</b> | malformed-header-sdp-b |
| <b>c=</b> | malformed-header-sdp-c |
| <b>i=</b> | malformed-header-sdp-i |
| <b>k=</b> | malformed-header-sdp-k |
| <b>m=</b> | malformed-header-sdp-m |
| <b>o=</b> | malformed-header-sdp-o |
| <b>r=</b> | malformed-header-sdp-r |
| <b>s=</b> | malformed-header-sdp-s |

| Attribute | VoIP profile option    |
|-----------|------------------------|
| <b>t=</b> | malformed-header-sdp-t |
| <b>v=</b> | malformed-header-sdp-v |
| <b>z=</b> | malformed-header-sdp-z |

## Discarding SIP messages with some malformed header and body lines

Enter the following command to configure deep SIP message inspection to discard SIP messages with a malformed Via line, a malformed route line or a malformed m= line but to pass messages with a malformed i= line or a malformed Max-Forwards line

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set malformed-header-via discard
      set malformed-header-route discard
      set malformed-header-sdp-m discard
      set malformed-header-sdp-i pass
      set malformed-header-max-forwards pass
    end
  end
```

## Discarding SIP messages with an unknown SIP message type

Enter the following command to discard SIP messages with an unknown SIP message line type as defined in all current SIP RFCs:

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set unknown-header discard
    end
  end
```

## Discarding SIP messages that exceed a message size

Enter the following command to set the maximum size of a SIP message to 200 bytes. Messages longer than 200 bytes are discarded.

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set max-body-length 200
    end
  end
```

The `max-body-length` option checks the value in the SIP Content-Length header line to determine body length. The Content-Length can be larger than the actual size of a SIP message if the SIP message content is split over more than one packet. SIP message sizes vary widely. The size of a SIP message can also change with the addition of Via and Record-Route headers as the message is transmitted between users and SIP servers.

## Discarding SIP messages with lines longer than 500 characters

Enter the following command to set the length of a SIP message line to 500 characters and to block messages that include lines with 500 or more characters:

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set max-line-length 500
      set block-long-lines enable
    end
  end
```

# Blocking SIP request messages

You may want to block different types of SIP requests:

- to prevent SIP attacks using these messages.
- If your SIP server cannot process some SIP messages because of a temporary issue (for example a bug that crashes or compromises the server when it receives a message of a certain type).
- Your SIP implementation does not use certain message types.

When you enable message blocking for a message type in a VoIP profile, whenever a security policy containing the VoIP profile accepts a SIP message of this type, the SIP ALG silently discards the message and records a log message about the action.

Use the following command to configure a VoIP profile to block SIP CANCEL and Update request messages:

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set block-cancel enable
      set block-update enable
    end
  end
```

SIP uses a variety of text-based messages or requests to communicate information about SIP clients and servers to the various components of the SIP network. Since SIP requests are simple text messages and since the requests or their replies can contain information about network components on either side of the FortiGate, it may be a security risk to allow these messages to pass through.

The following table lists all of the VoIP profile SIP request message blocking options. All of these options are disabled by default.



Blocking SIP OPTIONS messages may prevent a redundant configuration from operating correctly. See [Supporting geographic redundancy when blocking OPTIONS messages on page 115](#) for information about resolving this problem.

## Options for blocking SIP request messages

| SIP request message | SIP message blocking CLI Option |
|---------------------|---------------------------------|
| <b>ACK</b>          | block-ack                       |
| <b>BYE</b>          | block-bye                       |
| <b>Cancel</b>       | block-cancel                    |
| <b>INFO</b>         | block-info                      |
| <b>INVITE</b>       | block-invite                    |

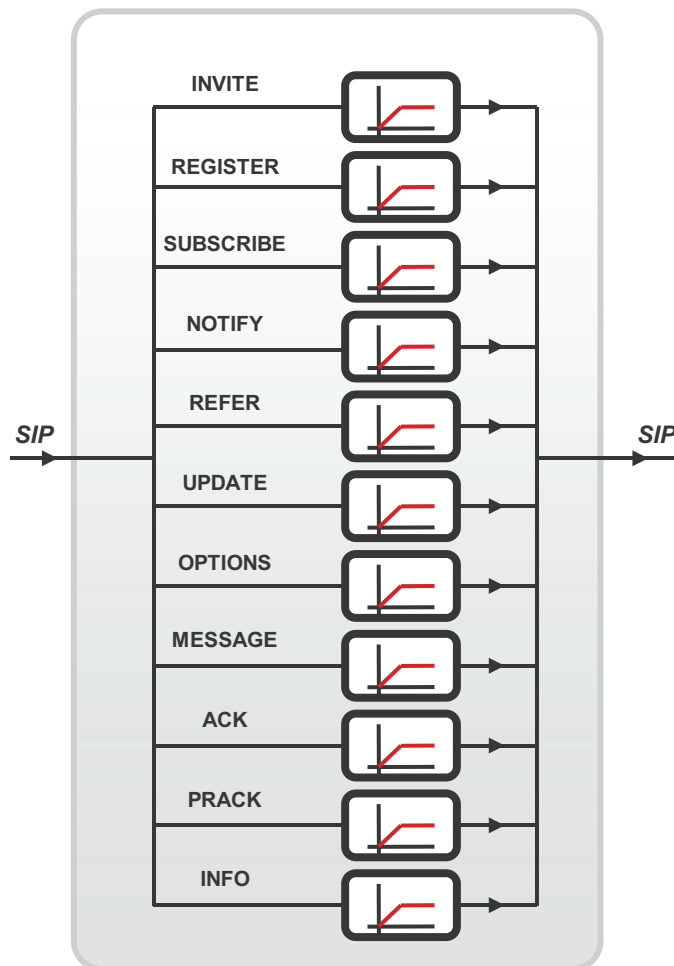


| SIP request message | SIP message blocking CLI Option |
|---------------------|---------------------------------|
| <b>Message</b>      | block-message                   |
| <b>Notify</b>       | block-notify                    |
| <b>Options</b>      | block-options                   |
| <b>PRACK</b>        | block-prack                     |
| <b>Publish</b>      | block-publish                   |
| <b>Refer</b>        | block-refer                     |
| <b>Register</b>     | block-register                  |
| <b>Subscribe</b>    | block-subscribe                 |
| <b>Update</b>       | block-update                    |

## SIP rate limiting

Configurable threshold for SIP message rates per request method. Protects SIP servers from SIP overload and DoS attacks.

### SIP rate limiting



FortiGates support rate limiting for the following types of VoIP traffic:

- Session Initiation Protocol (SIP)
- Skinny Call Control Protocol (SCCP) (most versions)
- Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE).

You can use rate limiting of these VoIP protocols to protect the FortiGate and your network from SIP and SCCP Denial of Service (DoS) attacks. Rate limiting protects against SIP DoS attacks by limiting the number of SIP REGISTER and INVITE requests that the FortiGate receives per second. Rate limiting protects against SCCP DoS attacks by limiting the number of SCCP call setup messages that the FortiGate receives per minute.

You configure rate limiting for a message type by specifying a limit for the number of messages that can be received per second. The rate is limited per security policy. When VoIP rate limiting is enabled for a message type, if the a single security policy accepts more messages per second than the configured rate, the extra messages are dropped and log messages are written when the messages are dropped.

Use the following command to configure a VoIP profile to limit the number of INVITE messages accepted by each security policy that the VoIP profile is added to 100 INVITE messages a second:

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set invite-rate 100
    end
  end
```

If you are experiencing denial of service attacks from traffic using these VoIP protocols, you can enable VoIP rate limiting and limit the rates for your network. Limit the rates depending on the amount of SIP and SCCP traffic that you expect the FortiGate to be handling. You can adjust the settings if some calls are lost or if the amount of SIP or SCCP traffic is affecting FortiGate performance.

The table below lists all of the VoIP profile SIP rate limiting options. All of these options are set to 0 so are disabled by default.



Blocking SIP OPTIONS messages may prevent a redundant configuration from operating correctly. See [Supporting geographic redundancy when blocking OPTIONS messages on page 115](#) for information about resolving this problem.

### Options for SIP rate limiting

| SIP request message | Rate Limiting CLI Option |
|---------------------|--------------------------|
| <b>ACK</b>          | ack-rate                 |
| <b>BYE</b>          | bye-rate                 |
| <b>Cancel</b>       | cancel-rate              |
| <b>INFO</b>         | info-rate                |
| <b>INVITE</b>       | invite-rate              |
| <b>Message</b>      | message-rate             |
| <b>Notify</b>       | notify-rate              |
| <b>Options</b>      | options-rate             |
| <b>PRACK</b>        | prack-rate               |
| <b>Publish</b>      | publish-rate             |

| SIP request message | Rate Limiting CLI Option |
|---------------------|--------------------------|
| <b>Refer</b>        | refer-rate               |
| <b>Register</b>     | register-rate            |
| <b>Subscribe</b>    | subscribe-rate           |
| <b>Update</b>       | update-rate              |

## Limiting the number of SIP dialogs accepted by a security policy

In addition to limiting the rates for receiving SIP messages, you can use the following command to limit the number of SIP dialogs (or SIP calls) that the FortiGate accepts.

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set max-dialogs 2000
    end
  end
```

This command sets the maximum number of SIP dialogs that can be open for SIP sessions accepted by any security policy that you add the VoIP profile to. The default setting of 0 does not limit the number of dialogs. You can add a limit to control the number of open dialogs and raise and lower it as required. You might want to limit the number of open dialogs for protection against SIP-based attackers opening large numbers of SIP dialogs. Every dialog takes memory and FortiGate CPU resources to process. Limiting the number of dialogs may improve the overall performance of the FortiGate. Limiting the number of dialogs will not drop calls in progress but may prevent new calls from connecting.

## SIP logging

You can enable SIP logging and logging of SIP violations in a VoIP profile.

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set log-call-summary enable
      set log-violations enable
    end
  end
```

To view SIP log messages go to **Log & Report > Forward Traffic**.

## Inspecting SIP over SSL/TLS (secure SIP)

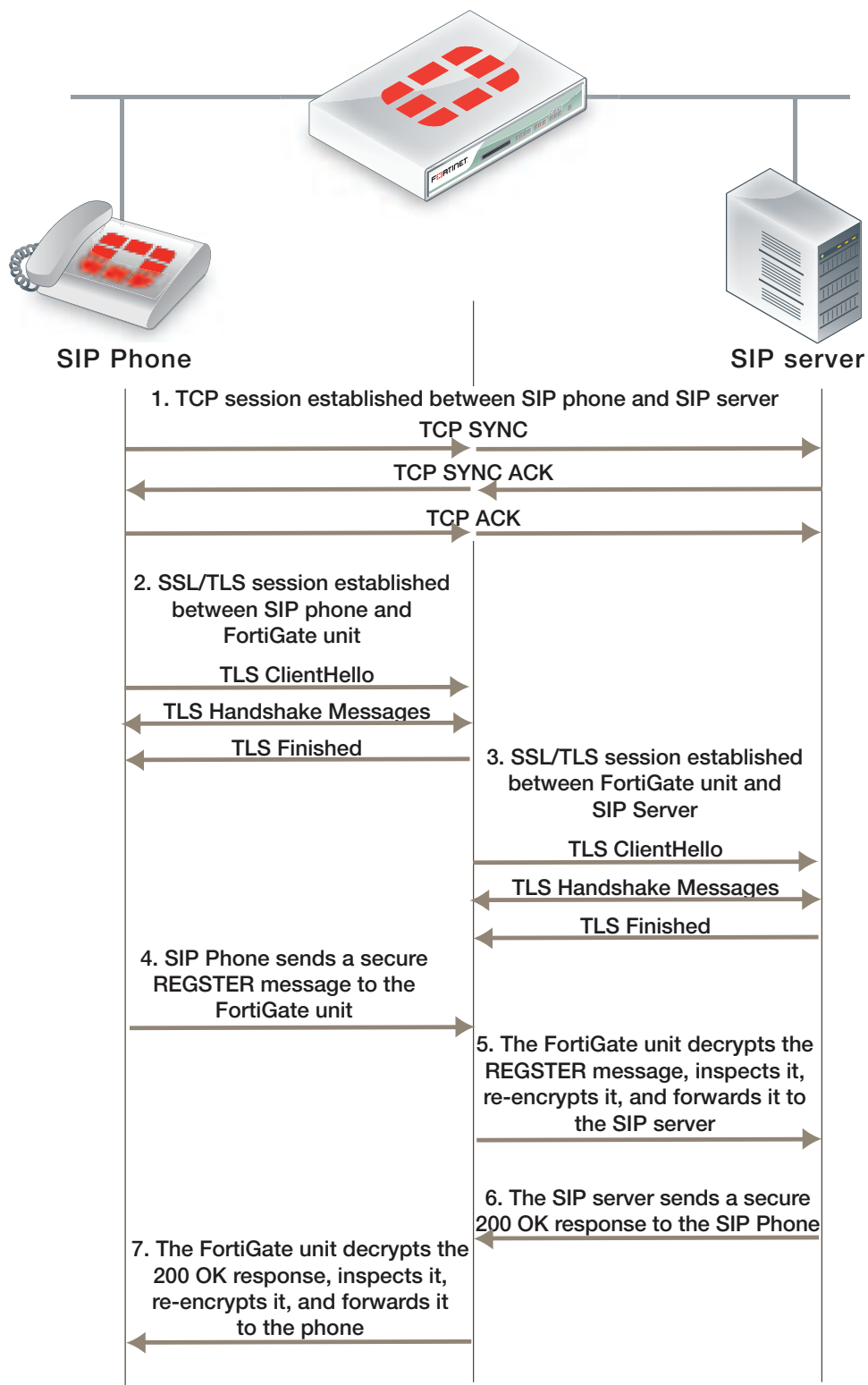
Some SIP phones and SIP servers can communicate using SSL or TLS to encrypt the SIP signaling traffic. To allow SIP over SSL/TLS calls to pass through the FortiGate, the encrypted signaling traffic has to be unencrypted and inspected. To do this, the FortiGate SIP ALG intercepts and unencrypts and inspects the SIP packets. The packets are then re-encrypted and forwarded to their destination.

Normally SIP over SSL/TLS uses port 5061. You can use the following command to change the port that the FortiGate listens on for SIP over SSL/TLS sessions to port 5066:

```
config system settings
    set sip-ssl-port 5066
end
```

The SIP ALG supports full mode SSL/TLS only. Traffic between SIP phones and the FortiGate and between the FortiGate and the SIP server is always encrypted.

You enable SSL/TLS SIP communication by enabling SSL mode in a VoIP profile. You also need to install the SIP server and client certificates on your FortiGate and add them to the SSL configuration in the VoIP profile.

**SIP over SSL/TLS between a SIP phone and a SIP server**

Other than enabling SSL mode and making sure the security policies accept the encrypted traffic, the FortiGate configuration for SSL/TLS SIP is the same as any SIP configuration. SIP over SSL/TLS is supported for all supported SIP configurations.

## Adding the SIP server and client certificates

A VoIP profile that supports SSL/TLS SIP requires one certification for the SIP server and one certificate that is used by all of the clients. Use the following steps to add these certificates to the FortiGate. Before you start, make sure the client and server certificate files and their key files are accessible from the management computer.

1. Go to **System > Certificates** and select **Import**.
2. Set **Type** to **Certificate**.
3. Browse to the **Certificate file** and the **Key file** and select **OK**.
4. Enter a password for the certificate and select **OK**.  
The certificate and key are uploaded to the FortiGate and added to the **Local Certificates** List.
5. Repeat to upload the other certificate.

The certificates are added to the list of Local Certificates as the filenames you uploaded. You can add comments to make it clear where its from and how it is intended to be used.

## Adding SIP over SSL/TLS support to a VoIP profile

Use the following commands to add SIP over SSL/TLS support to the default VoIP profile. The following command enables SSL mode and adds the client and server certificates and passwords, the same ones you entered when you imported the certificates:

```
config voip profile
  edit default
    config sip
      set ssl-mode full
      set ssl-client-certificate "Client_cert"
      set ssl-server-certificate "Server_cert"
      set ssl-auth-client "check-server"
      set ssl-auth-server "check-server-group"
    end
  end
```

Other SSL mode options are also available:

|   |  |
|---|--|
| <code>ssl-send-empty-frags {disable   enable}</code>          | Enable to send empty fragments to avoid CBC IV attacks. Compatible with SSL 3.0 and TLS 1.0 only. Default is <code>enable</code> .   |
| <code>ssl-client-renegotiation {allow   deny   secure}</code> | Control how the ALG responds when a client attempts to renegotiate the SSL session. You can allow renegotiation or block sessions when the client attempts to renegotiate. You can also select <code>secure</code> to reject an SSL connection that does not support <a href="#">RFC 5746</a> secure renegotiation indication. Default is <code>allow</code> . |



|  |   |
|--|---|
| <code>ssl-algorithm {high   low   medium}</code>           | Select the relative strength of the algorithms that can be selected. You can select <code>high</code> , the default, to allow only AES or 3DES, <code>medium</code> , to allow AES, 3DES, or RC4 or <code>low</code> , to allow AES, 3DES, RC4, or DES. |
| <code>ssl-pfs {allow   deny   require}</code>              | Select whether to <code>allow</code> , <code>deny</code> , or <code>require</code> perfect forward secrecy (PFS). Default is <code>allow</code> .   |
| <code>ssl-min-version {ssl-3.0   tls-1.0   tls-1.1}</code> | Select the minimum level of SSL support to allow. The default is <code>ssl-3.0</code> .   |
| <code>ssl-max-version {ssl-3.0   tls-1.0   tls-1.1}</code> | Select the maximum level of SSL support to allow. The default is <code>tls-1.1</code> .   |

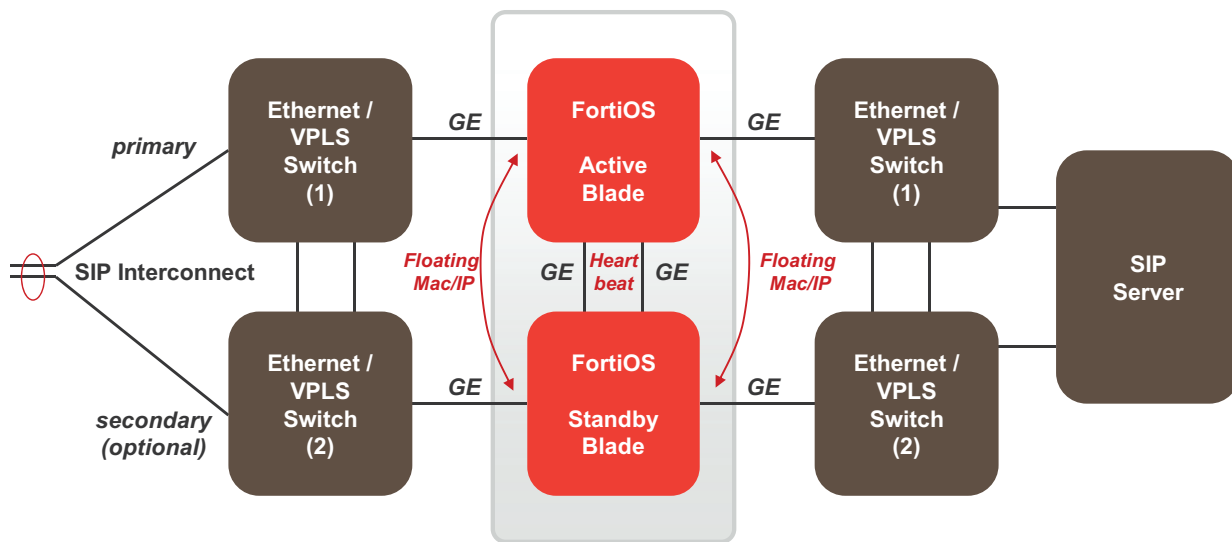
## SIP and HA-session failover and geographic redundancy

FortiGate high availability supports SIP session failover (also called stateful failover) for active-passive HA. To support SIP session failover, create a standard HA configuration and select the Enable Session Pick-up option.

SIP session failover replicates SIP states to all cluster units. If an HA failover occurs, all in progress SIP calls (setup complete) and their RTP flows are maintained and the calls will continue after the failover with minimal or no interruption.

SIP calls being set up at the time of a failover may lose signaling messages. In most cases the SIP clients and servers should use message retransmission to complete the call setup after the failover has completed. As a result, SIP users may experience a delay if their calls are being set up when an HA failover occurs. But in most cases the call setup should be able to continue after the failover.

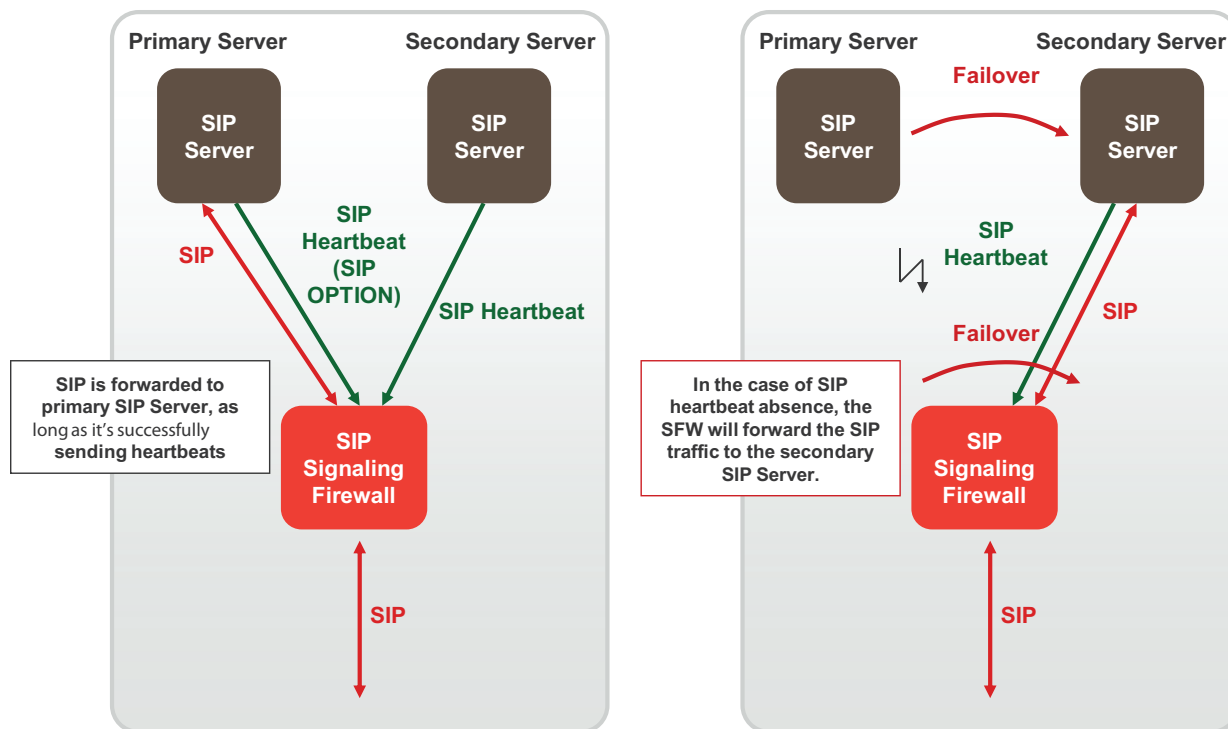
### SIP HA session failover



In some cases, failover during call teardown can result in hanging RTP connections which can accumulate over time and use up system memory. If this becomes a problem, you can set a time for the `call-keepalive` SIP VoIP profile setting. The FortiGate will then terminate calls with no activity after the time limit is exceeded. Range is 1 to 10,080 seconds. This options should be used with caution because it results in extra FortiGate CPU overhead and can cause delay/jitter for the VoIP call. Also, the FortiGate terminates the call without sending SIP messages to end the call. And if the SIP endpoints send SIP messages to terminate the call they will be blocked by the FortiGate if they are sent after the FortiGate terminates the call.

## SIP geographic redundancy

Maintains a active-standby SIP server configuration, which even supports geographical distribution. If the active SIP server fails (missing SIP heartbeat messages or SIP traffic) FortiOS will redirect the SIP traffic to a secondary SIP server. No special configuration is required for geographic redundancy, just standard HA configuration.

**Geographic redundancy****Supporting geographic redundancy when blocking OPTIONS messages**

For some geographic redundant SIP configurations, the SIP servers may use SIP OPTIONS messages as heartbeats to notify the FortiGate that they are still operating (or alive). This is a kind of passive SIP monitoring mechanism where the FortiGate isn't actively monitoring the SIP servers and instead the FortiGate passively receives and analyzes OPTIONS messages from the SIP servers.

If FortiGates block SIP OPTIONS messages because `block-options` is enabled, the configuration may fail to operate correctly because the OPTIONS messages are blocked by one or more FortiGates.

However, you can work around this problem by enabling the `block-geo-red-options` application control list option. This option causes the FortiGate to refresh the local SIP server status when it receives an OPTIONS message before dropping the message. The end result is the heartbeat signals between geographically redundant SIP servers are maintained but OPTIONS messages do not pass through the FortiGate.

Use the following command to block OPTIONS messages while still supporting geographic redundancy:

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set block-options disable
      set block-geo-red-options enable
    end
  end
```



The `block-options` option setting overrides the `block-geo-red-options` option. If `block-options` is enabled the FortiGate only blocks SIP OPTIONS messages and does not refresh local SIP server status.

---

## Support for RFC 2543-compliant branch parameters

RFC 3261 is the most recent SIP RFC, it obsoletes RFC 2543. However, some SIP implementations may use RFC 2543-compliant SIP calls.

The `rfc2543-branch` VoIP profile option allows the FortiGate to support SIP calls that include an RFC 2543-compliant branch parameter in the SIP Via header. This option also allows FortiGates to support SIP calls that include Via headers that are missing the branch parameter.

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set rfc2543-branch enable
    end
  end
```

## SIP and IPS

You can enable IPS in security policies that also accept SIP sessions to protect the SIP traffic from SIP-based attacks. If you enable IPS in this way then by default the pinholes that the SIP ALG creates to allow RTP and RTCP to flow through the firewall will also have IPS enabled.

This inheritance of the IPS setting can cause performance problems if the RTP traffic volume is high since IPS checking may reduce performance in some cases. Also if you are using network processor (NP) interfaces to accelerate VoIP performance, when IPS is enabled for the pinhole traffic is diverted to the IPS and as a result is not accelerated by the network processors.

You can use the following CLI command to disable IPS for the RTP pinhole traffic.

```
config voip profile
  edit VoIP_Pro_Name
    config sip
      set ips-rtp disable
    end
  end
```

# SIP debugging

This chapter includes some information to help with debugging SIP configurations.

## SIP debug log format

Assuming that `diagnose debug console timestamp` is enabled then the following shows the debug that is generated for an INVITE if `diag debug appl sip -1` is enabled:

```
2010-01-04 21:39:59 sip port 26 locate session for 192.168.2.134:5061 ->
172.16.67.192:5060
2010-01-04 21:39:59 sip sess 0x979df38 found for 192.168.2.134:5061 ->
172.16.67.192:5060
2010-01-04 21:39:59 sip port 26 192.168.2.134:5061 -> 172.16.67.192:5060
2010-01-04 21:39:59 sip port 26 read [(0,515)]
(494e56495445207369703a73657276696365403139322e3136382e322e3130303a35303630205349502f322e300d0a5669613a2
05349502f322e302f554450203132372e302e312e313a353036313b6272616e63683d7a39684734624b2d363832372d3632302d3
00d0a46726f6d3a2073697070203c7369703a73697070403132372e302e312e313a353036313e3b7461673d36383237534950705
4616730303632300d0a546f3a20737574203c7369703a73657276696365403139322e3136382e322e3130303a353036303e0d0a4
3616c6c2d49443a203632302d36383237403132372e302e312e310d0a435365713a203120494e564954450d0a436f6e746163743
a207369703a73697070403132372e302e312e313a353036310d0a4d61782d466f7277617264733a2037300d0a5375626a6563743
a20506572666f726d616e636520546573740d0a436f6e74656e742d547970653a206170706c69636174696f6e2f7364700d0a436
f6e74656e742d4c656e6774683a20203132390d0a0d0a763d300d0a6f3d75736572312035333635337363520323335333638373
6333720494e20495034203132372e302e312e310d0a733d2d0d0a633d494e20495034203132372e302e312e310d0a743d3020300
d0a6d3d617564696f2036303031205254502f41565020300d0a613d7274706d61703a302050434d552f383030300d0a) (INVITE
sip:service@192.168.2.100:5060 SIP/2.0..Via: SIP/2.0/UDP
127.0.1.1:5061;branch=z9hG4bK-6827-620-0..From: sipp
%lt;sip:sipp@127.0.1.1:5061>;tag=6827SIPpTag00620..To: sut
%lt;sip:service@192.168.2.100:5060>..Call-ID: 620-6827@127.0.1.1..CSeq: 1
INVITE..Contact: sip:sipp@127.0.1.1:5061..Max-Forwards: 70..Subject: Performance
Test..Content-Type: application/sdp..Content-Length: 129...v=0..o=user1 53655765
2353687637 IN IP4 127.0.1.1..s=-..c=IN IP4 127.0.1.1..t=0 0..m=audio 6001 RTP/AVP
0..a=rtpmap:0 PCMU/8000..)
2010-01-04 21:39:59 sip port 26 len 515
2010-01-04 21:39:59 sip port 26 INVITE '192.168.2.100:5060' addr 192.168.2.100:5060
2010-01-04 21:39:59 sip port 26 CSeq: 1 INVITE
2010-01-04 21:39:59 sip port 26 Via: UDP 127.0.1.1:5061 len 14 received 0 rport 0 0 branch 'z9hG4bK-
6827-620-0'
2010-01-04 21:39:59 sip port 26 From: 'sipp ;tag=6827SIPpTag00620' URI 'sip:sipp@127.0.1.1:5061' tag
'6827SIPpTag00620'
2010-01-04 21:39:59 sip port 26 To: 'sut ' URI 'sip:service@192.168.2.100:5060' tag ''
2010-01-04 21:39:59 sip port 26 Call-ID: '620-6827@127.0.1.1'
2010-01-04 21:39:59 sip port 26 Contact: '127.0.1.1:5061' addr 127.0.1.1:5061 expires 0
2010-01-04 21:39:59 sip port 26 Content-Length: 129 len 3
2010-01-04 21:39:59 sip port 26 sdp o=127.0.1.1 len=9
2010-01-04 21:39:59 sip port 26 sdp c=127.0.1.1 len=9
2010-01-04 21:39:59 sip port 26 sdp m=6001 len=4
2010-01-04 21:39:59 sip port 26 find call 0 '620-6827@127.0.1.1'
2010-01-04 21:39:59 sip port 26 not found
2010-01-04 21:39:59 sip port 26 call 0x97a47c0 open (collision (nil))
2010-01-04 21:39:59 sip port 26 call 0x97a47c0 open txn 0x979f7f8 INVITE dir 0
2010-01-04 21:39:59 sip port 26 sdp i: 127.0.1.1:6001
2010-01-04 21:39:59 sip port 26 policy id 1 is_client_vs_policy 1 policy_dir_rev 0
2010-01-04 21:39:59 sip port 26 policy 1 not RTP policy
2010-01-04 21:39:59 sip port 26 learn sdp from stream address
2010-01-04 21:39:59 sip port 26 call 0x97a47c0 sdp 172.16.67.198:43722
2010-01-04 21:39:59 sip port 26 call 0x97a47c0 txn 0x979f7f8 127.0.1.1:5061 find new address and port
2010-01-04 21:39:59 sip port 26 call 0x97a47c0 txn 0x979f7f8 127.0.1.1:5061 find new address and port
2010-01-04 21:39:59 sip port 26 call 0x97a47c0 txn 0x979f7f8 127.0.1.1:5061 find new address and port
2010-01-04 21:39:59 sip port 30 write 192.168.2.134:5061 -> 172.16.67.192:5060 (13,539)
2010-01-04 21:39:59 sip port 30 write [(13,539)]
```

```
(494e56495445207369703a73657276696365403137322e31362e36372e3139323a35303630205349502f322e300d0a5669613a2
05349502f322e302f554450203137322e31362e36372e3139383a35323036353b6272616e63683d7a39684734624b2d363832372
d3632302d300d0a46726f6d3a2073697070203c7369703a73697070403137322e31362e36372e3139383a34333732343e3b74616
73d363832375349507054616730303632300d0a546f3a20737574203c7369703a73657276696365403137322e31362e36372e313
9323a353036303e0d0a43616c6c2d49443a203632302d36383237403132372e302e312e310d0a435365713a203120494e5649544
50d0a436f6e746163743a207369703a73697070403137322e31362e36372e3139383a34333732350d0a4d61782d466f727761726
4733a2037300d0a5375626a6563743a20506572666f726d616e636520546573740d0a436f6e74656e742d547970653a206170706
c69636174696f6e2f7364700d0a436f6e74656e742d4c656e6774683a20203133380d0a0d0a763d300d0a6f3d757365723120353
336353373635203233353336383736333720494e20495034203137322e31362e36372e3139380d0a733d2d0d0a633d494e20495
034203137322e31362e36372e3139380d0a743d3020300d0a6d3d617564696f203433373232205254502f41565020300d0a613d7
274706d61703a302050434d552f383030300d0a) (INVITE sip:service@172.16.67.192:5060 SIP/2.0..Via: SIP/2.0/UDP
172.16.67.198:52065;branch=z9hG4bK-6827-620-0..From: sipp ;tag=6827SIPpTag00620..To: sut ..Call-ID: 620-
6827@127.0.1.1..CSeq: 1 INVITE..Contact: sip:sipp@172.16.67.198:43725..Max-Forwards: 70..Subject:
Performance Test..Content-Type: application/sdp..Content-Length: 138....v=0..o=user1 53655765 2353687637
IN IP4 172.16.67.198..s=-..c=IN IP4 172.16.67.198..t=0 0..m=audio 43722 RTP/AVP 0..a=rtpmap:0
PCMU/8000..)]
```

## SIP-proxy filter per VDOM

You can use the `diagnose sys sip-proxy xxx` command in a VDOM to get info about how SIP is operating in each VDOM.

## SIP-proxy filter command

Use the `diagnose system sip-proxy filter` to filter diagnose information for the SIP ALG. The following filters are available:

```
diag sys sip-proxy filter vd
diag sys sip-proxy filter dst-addr4
diag sys sip-proxy filter dst-addr6
diag sys sip-proxy filter dst-port
diag sys sip-proxy filter identity-policy
diag sys sip-proxy filter negate
diag sys sip-proxy filter policy
diag sys sip-proxy filter policy-type
diag sys sip-proxy filter profile-group
diag sys sip-proxy filter src-addr4
diag sys sip-proxy filter src-addr6
diag sys sip-proxy filter src-port
diag sys sip-proxy filter vd
diag sys sip-proxy filter voip-profile
```

You can clear, view and negate/invert the sense of a filter using these commands:

```
diag sys sip-proxy filter clear
diag sys sip-proxy filter list
diag sys sip-proxy filter negate
```

## SIP debug setting

Control of the SIP debug output is governed by the following command

```
diagnose debug application sip <debug_level_int>
```

Where the `<debug_level_int>` is a bitmask and the individual values determine whether the listed items are logged or not. The `<debug_level_int>` can be:

|             |  |
|-------------|--|
| <b>1</b>    | Configuration changes, mainly addition/deletion/modification of virtual domains.   |
| <b>2</b>    | TCP connection accepts or connects, redirect creation.   |
| <b>4</b>    | Create or delete a session.  |
| <b>16</b>   | Any IO read or write.  |
| <b>32</b>   | An ASCII dump of all data read or written.   |
| <b>64</b>   | Include HEX dump in the above output.  |
| <b>128</b>  | Any activity related to the use of the FortiCarrier dynamic profile feature to determine the correct profile-group to use. |
| <b>256</b>  | Log summary of interesting fields in a SIP call.   |
| <b>1024</b> | Any activity related to SIP geo-redundancy.  |
| <b>2048</b> | Any activity related to HA syncing of SIP calls.   |

## Display SIP rate-limit data

You can use the `diagnose sys sip-proxy meters` command to display SIP rate limiting data.

For the following command output `rate 1` shows that the current (over last second) measured rate for INVITE/ACK and BYE was 1 per second, the `peak 1` shows that the peak rate recorded is 1 per second, the `max 0` shows that there is no maximum limit set, the `count 18` indicates that 18 messages were received and `drop 0` indicates that none were dropped due to being over the limit.

```
diag sys sip-proxy meters
sip
sip vd: 0
sip policy: 1
sip identity-policy: 0
sip policy-type: IPv4
sip profile-group:
sip dialogs: 18
sip dialog-limit: 0
sip UNKNOWN: rate 0 peak 0 max 0 count 0 drop 0
sip ACK: rate 1 peak 1 max 0 count 18 drop 0
sip BYE: rate 1 peak 1 max 0 count 18 drop 0
sip CANCEL: rate 0 peak 0 max 0 count 0 drop 0
sip INFO: rate 0 peak 0 max 0 count 0 drop 0
sip INVITE: rate 1 peak 1 max 0 count 18 drop 0
sip MESSAGE: rate 0 peak 0 max 0 count 0 drop 0
sip NOTIFY: rate 0 peak 0 max 0 count 0 drop 0
sip OPTIONS: rate 0 peak 0 max 0 count 0 drop 0
sip PRACK: rate 0 peak 0 max 0 count 0 drop 0
```



```
sip PUBLISH: rate 0 peak 0 max 0 count 0 drop 0
sip REFER: rate 0 peak 0 max 0 count 0 drop 0
sip REGISTER: rate 0 peak 0 max 0 count 0 drop 0
sip SUBSCRIBE: rate 0 peak 0 max 0 count 0 drop 0
sip UPDATE: rate 0 peak 0 max 0 count 0 drop 0
sip PING: rate 0 peak 0 max 0 count 0 drop 0
sip YAHOOREF: rate 0 peak 0 max 0 count 0 drop 0
```



**FORTINET®**

*High Performance Network Security*



Copyright© 2017 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., in the U.S. and other jurisdictions, and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's General Counsel, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. In no event does Fortinet make any commitment related to future deliverables, features, or development, and circumstances may change such that any forward-looking statements herein are not accurate. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.